AC ELECTRONICS DIVISION GENERAL MOTORS CORPORATION

INTERPLANETARY GUIDANCE SYSTEM REQUIREMENTS STUDY

VOLUME II

COMPUTER PROGRAM DESCRIPTIONS

PART 2

PERFORMANCE ASSESSMENT OF MIDCOURSE GUIDANCE SYSTEMS

Prepared for
ERC Systems Laboratory
under Contract NAS 12-7

AC ELECTRONICS DIVISION
General Motors Corporation
El Segundo, California

15 September 1966

TABLE OF CONTENTS

PARAGRAPH	TITLE	PAGE
ABSTRACT		vii
1.0	INTRODUCTION AND SUMMARY	1-1
2.0	MATHEMATICAL MODEL	2-1
	2.1 TRAJECTORY	2-1
	2.1.1 Nominal Trajectory	2-1
	2.1.2 Actual Trajectory	2-10
	2.2 SENSORS	2-11
	2.2.1 Space Sextant	2-11
	2.2.2 Horizon Sensor	2-14
	2.2.3 Planet Tracker	2-14
	2.2.4 Ground Tracking	2-15
	2.3 NAVIGATION	2-18
	2.3.1 The State Vector	2-19
	2.3.2 Orbit Rectification	2-20
	2.3.3 Square-root Kalman Filter	2-21
	2.3.4 Compensation for Computational Errors	2-21
	2.3.5 Variable Computer Word-length	2-21
	2.4 GUIDANCE	2-21
	2.4.1 Determination of the Desired Velocity Correction	2-22
	2.4.2 Timing of the Velocity Corrections	2-23
	2.4.3 Implementation of the Correction	2-24
	2.4.4 Relation to the General Guidance Problem	2-25
3.0	COMPUTER PROGRAM DESCRIPTION	3-1
	3.1 INTRODUCTORY AND EXPLANATORY REMARKS	3-1
	3.1.1 Schema for Flow Chart Presentation	3-2
	3.1.2 Definition of Flow Chart Symbols	3-3
	3.1.3 Definition of Nomenclature	3-5
	3.2 FUNCTIONAL ORGANIZATION OF THE PROGRAM	3-26
	3.3 INPUT, GENERAL INITIALIZATION AND OUTPUT	3-29
	3.3.1 Input - Block A	3-29
	3.3.2 General Initialization - Block B	3-44
	3.3.3 Output - Block C	3-63

TABLE OF CONTENTS

PARAGRAPH	TITLE	PAGE
3.4	BASIC COMPUTATIONAL BLOCKS	3-92
3.4.1	Two-Body Interplanetary Nominal - Block I	3-92
3.4.2	Explicit State Transition - Block II	3-106
3.4.3	Two-Body Actual - Block III	3-121
3.4.4	Guidance - Block IV	3-129
3.4.5	Electromagnetic Sensors - Block V	3-147
3.4.6	Navigation - Block VI	3-172
4.0	USER'S GUIDE: PROGRAM 284	4-1
4.1	INTRODUCTION	4-1
4.2	INPUT SHEETS AND GENERAL COMMENTS	4-2
4.2.1	Setup and Handling of the Input Tables	4-3
4.2.2	Stacking of Simulation Runs	4-5
4.2.3	Precautions	4-5
4.2.4	Input Sheets	4-5
4.3	TRAJECTORY AND STATE TRANSITION BLOCK SIMULATION	4-18
4.3.1	Nominal Block and State Transition Block	4-18
4.3.2	Actual Block Simulation	4-20
4.3.3	Precautions	4-20
4.4	ELECTROMAGNETIC SENSOR SIMULATION	4-21
4.4.1	How to Simulate Ground Trackers as the Aiding Instruments	4-21
4.4.2	How to Simulate the Horizon Sensor as the Aiding Instrument	4-26
4.4.3	How to Simulate the Space Sextant as the Aiding Instrument	4-29
4.5	NAVIGATION SIMULATION	4-33
4.5.1	Kalman Filter	4-33
4.5.2	Masking	4-34
4.5.3	Orbit Rectification	4-34
4.5.4	A Priori Statistics Error	4-34
4.5.5	"Fudge Factor" to Account for the Computational and Dynamic Model Errors	4-34
4.5.6	Precautions	4-35

TABLE OF CONTENTS

PARAGRAPH	TITLE	PAGE
	4.6 GUIDANCE SIMULATION	4-35
	4.6.1 Precautions	4-36
	4.6.2 Sample Input	4-36
	4.7 OUTPUT	4-36
	4.7.1 Output Computations	4-37
	4.7.2 Precautions	4-37
	4.7.3 Sample Input	4-38
	4.8 UNSATISFACTORY PROGRAM OPTIONS	4-38
5.0	OPERATOR'S GUIDE: PROGRAM 284	5-1
	5.1 GENERAL INFORMATION	5-1
	5.2 MACHINE CONFIGURATION	5-1
	5.3 TAPE USAGE	5-1
	5.4 DECK ARRANGEMENT	5-3
	5.4.1 Input Section	5-3
	5.4.2 Calculation Section	5-4
	5.4.3 Output Section (BLOCK)	5-4
	5.5 ERROR EXITS	5-7
	5.5.1 Input Card Errors	5-7
	5.5.2 Errors Occurring During Calculation	5-8
	5.5.3 Abnormal Errors	5-10
6.0	PROGRAMMER'S GUIDE: PROGRAM 284	6-1
	6.1 INTRODUCTION	6-1
	6.2 SUBROUTINE CONTROL CHARTS	6-6
	6.2.1 Main Program (284) Subroutine Control Chart	6-6
	6.3 SUBROUTINE DESCRIPTIONS	6-16
	6.4 SUBROUTINE FLOW CHARTS	6-26
	6.4.1 INPUTC - Input Section Control Program	6-27
	6.4.2 IMOVE - Input Section Preparation Subroutine	6-29
	6.4.3 CALCC - Calculate Section Control Program	6-35
	6.4.4 OUTPUT - Output Section Control Program	6-43
7.0	REFERENCES	7-1
APPENDIX A - PROGRAM LISTING, PROGRAM 284		A-1

ABSTRACT

This document describes the digital computer program that has been developed to study the guidance and navigation requirements for interplanetary missions. The presentation contains the structure of the mathematical model, a computer program description, and a user's guide. The program description is composed primarily of a hierarchical structure of flow charts with concomitant explanatory remarks. The mathematical model equations which are programmed into the computer are stated within the lowest-level flow charts. The flow charts are designed to provide insight into the program operations at several levels of logical and computational complexity. The program is coded in Fortran IV and a listing of the program is given in the Appendix.

1.0 INTRODUCTION AND SUMMARY

This document describes the structure of a digital computer simulation program for the performance assessment of midcourse guidance systems. The program can be used to study and assess the performance of space guidance systems during free-fall phases of interplanetary missions. Program 284 is organized in such a manner that a desired system configuration involving different combinations of aiding instruments can be easily simulated. Thus, comparative studies using ground based and/or on-board aiding instruments, as well as different sensor accuracies, can be performed.

The aiding instruments presently programmed in 284 are: three ground trackers capable of simulating the measurement of range, range rate, azimuth and elevation angles; a horizon sensor capable of simulating the on-board measurement of the subtended angle and the two local vertical angles; a planet tracker, i.e., a horizon sensor capable of operating in a planet tracker mode; and a space sextant capable of simulating the measurement of a planet's limb-to-star angle or the planet's center-to-star angle. The space sextant has the capability of using different star selection options including realistic stars selected from a star catalog in an "optimum" fashion.

The functional organization of the program is in a block structure. Therefore, additional aiding instruments could very easily be incorporated in the program if it is desired.

Linear guidance laws that incorporate different end constraints with fixed and variable time of arrival can be simulated in this program. Five different guidance laws are available. They cover the possibility to control the terminal position vector or terminal altitude and velocity direction (for both fixed and variable time of arrival), as well as the possibility to control altitude, flight path angle and trajectory plane for variable time of arrival.

The general performance assessment problem for space guidance systems is delineated in Volume I. In Paragraph 2.0 of this volume, supplemental information for the mathematical model and system configuration are described.

Paragraph 3.0 contains the detailed computer program description. This description is composed primarily of a hierarchial structure of flow charts, explanatory remarks and necessary equations. The flow charts are designed to provide insight into the program operations at several levels of logical and computational complexity. The "highest" level chart, designated as Level I, depicts the overall structure of the program. Each block appearing in this chart is then described by a lower-level chart, designated as Level II. This policy is repeated in every level until no further logic remains to be described. The equations which are programmed into the computer are normally stated in the lowest-level flow charts.

Paragraph 4.0 contains a description of the user's guide. In the user's guide the specific capabilities and options contained in the program are described. A set of input sheets are given and their use outlined. Illustrative examples using the various program options are also included.

Paragraph 5.0 contains an operator's guide. In this paragraph the machine configuration, tape assignments, deck arrangements and error exits are described.

Paragraph 6.0 contains a programmer's guide which indicates the correspondence between subroutines and sub-blocks defined in Paragraph 3.0, and which provides additional programming information for those sections of the program not adequately defined in Paragraph 3.0. Particular attention is given to those sections of the program which are of a control nature.

2.0 MATHEMATICAL MODEL

In this paragraph the basic equations necessary to specify the functions and assess the performance of a space guidance system during free-fall phases of interplanetary missions are presented. The assumptions upon which the equations are based are stated and justified in Volume I.

The free-fall phases include planetary departure and planetary approach phases, transfer between planets, and planetocentric free-fall orbits. A free-fall phase is defined as the part of the mission in which the position and velocity of the space vehicle is affected by gravitational forces only. That means the mathematical model does not describe the performance during a trajectory change maneuver as dictated by the nominal trajectory design. It does, however, include the effect of midcourse corrections upon the performance of the system.

2.1 TRAJECTORY

2.1.1 Nominal Trajectory

There are two conceptually different methods to determine the nominal free-fall trajectory of an interplanetary mission. The first method uses the "exact" dynamical equations; i.e., the influence of all heavenly bodies upon the space vehicle are taken into account at each time during the entire mission. In this case, no "closed-form" analytical solutions exist and the analyst has to rely solely upon numerical integration. The second method is an approximate one. It utilizes the fact that the vehicle's motion in different portions of free-fall is essentially determined by the gravitational attraction of a single central body. That means the motion can be described by different two-body orbits in different portions of the flight, namely, heliocentric ellipses determined by the initial conditions and the sun in the transfer phase, and planetocentric ellipses or hyperbolas during planet approach, planet departure phase, and periodic orbits around a planet. If the different two-body orbits are appropriately "matched," the resulting orbit constitutes, as experience has shown, an excellent approximation to the precise orbit. This method is employed here in the mathematical model. Since the differential equations for the two-body problem can be solved in closed form, nominal position and velocity at a desired time are obtained from explicit analytical expressions rather than by numerical integration. The steps involved are the following:

a. At the beginning of the m-th conic, position and velocity are obtained in an equatorial coordinate system. They are defined by the symbols

$$\underline{R}^*(t_o^m) \overset{Df}{=} \begin{pmatrix} X_1^*(t_o^m) \\ X_2^*(t_o^m) \\ X_3^*(t_o^m) \end{pmatrix}; \qquad \underline{V}^*(t_o^m) \overset{Df}{=} \begin{pmatrix} X_4^*(t_o^m) \\ X_5^*(t_o^m) \\ X_6^*(t_o^m) \end{pmatrix}$$

b. The corresponding angular momentum

$$\underline{h}^*(t_o^m) = \underline{R}^*(t_o^m) \times \underline{V}^*(t_o^m)$$

is computed.

c. From the vis-viva

$$\alpha^*(t_o^m) = \frac{2}{R^*(t_o^m)} - \frac{\underline{V}^*(t_o^m) \cdot \underline{V}^*(t_o^m)}{\mu_\ell}$$

is computed, where μ_ℓ is the gravitational constant of the body ℓ. The sign of this quantity defines the character of the orbit. For closed orbits $\alpha^*(t_o^m)$ is positive; for hyperbolic orbit, it becomes negative.

d. Unit vectors ($\underline{\zeta}_1$, $\underline{\zeta}_2$, $\underline{\zeta}_3$) in the direction of $\underline{R}^*$ (we drop t_o^m in the following equations for the sake of brevity) $\underline{h}^*$ and $\underline{h}^* \times \underline{R}^*$ define the local coordinate system and the orientation of the orbital plane at t_o^m; i.e.,

$$\underline{\zeta}_1 = \frac{\underline{R}^*}{R}; \quad \underline{\zeta}_2 = \underline{\zeta}_3 \times \underline{\zeta}_1; \quad \underline{\zeta}_3 = \frac{\underline{R}^* \times \underline{V}^*}{|\underline{R}^* \times \underline{V}^*|}$$

A second, right-handed, planet-centered, orbital in-plane coordinate system (ξ_1, ξ_2, ξ_3) is defined such that $\underline{\xi}_1$ points along the pericentron and $\underline{\xi}_3$ is perpendicular to the trajectory plane; i.e.,

$$\underline{\xi}_3 \overset{Df}{=} \underline{\zeta}_3$$

The transformation between the (ξ_1, ξ_2, ξ_3) and the (X_1, X_2, X_3) coordinates at $t = t_k^m$ is given by

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}_{t_k^m} = \mathcal{R}^*(t_o^m) \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \equiv 0 \end{pmatrix}_{t_k^m}$$

where

$$\mathcal{R}^*(t_o^m) = (\zeta_1\ \zeta_2\ \zeta_3)_{t_o^m} \begin{pmatrix} \cos\nu & \sin\nu & 0 \\ -\sin\nu & \cos\nu & 0 \\ 0 & 0 & 1 \end{pmatrix}_{t_c^m}$$

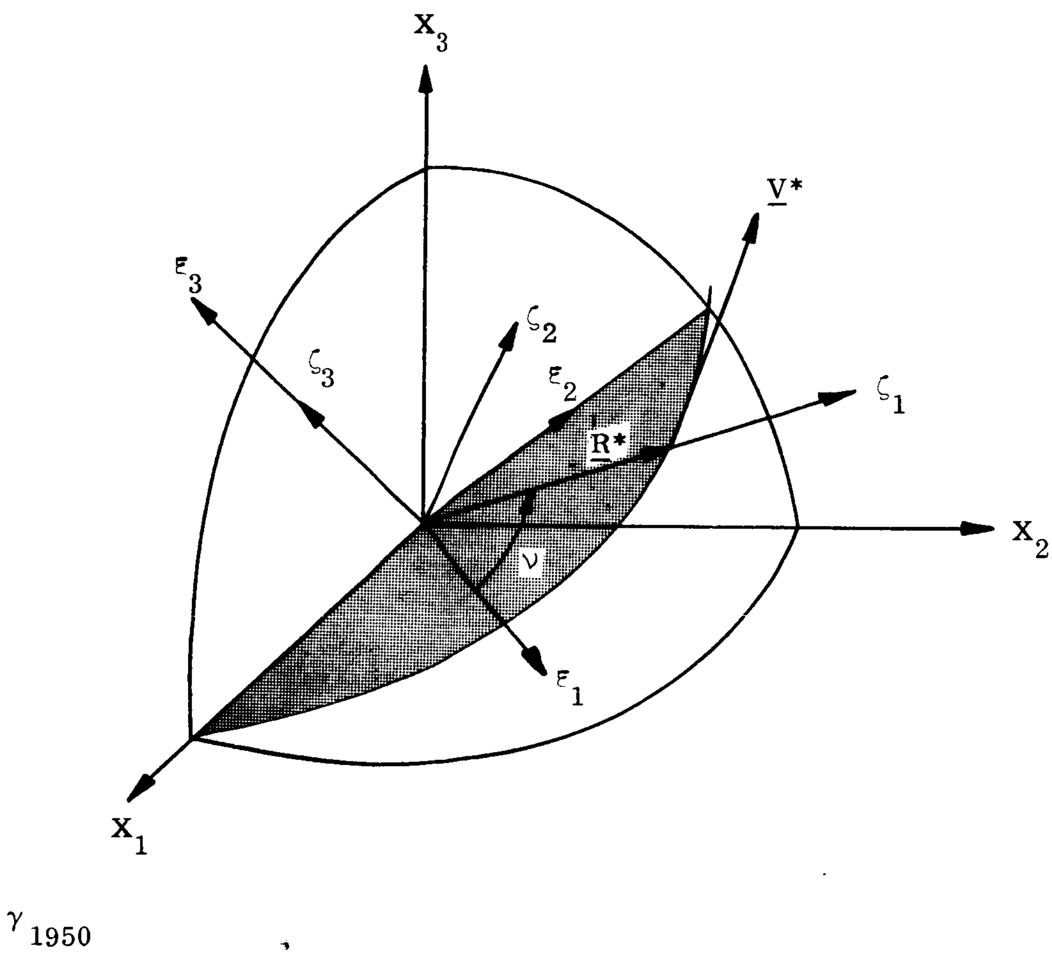


Figure 2. Definition of Coordinate Systems

e. In this orbital plane, the trajectory is specified in the case of an elliptical orbit in the following way:

$$\alpha^* > 0 \Rightarrow \text{elliptic case}$$

The semi-major axis is given by

$$a = \frac{1}{\alpha}$$

The eccentricity is given by

$$e = \{(1 - \frac{R}{a})^2 + \frac{[\underline{R} \cdot \underline{V}]^2}{\mu_\ell \, a}\}^{1/2}$$

The eccentric anomaly can be expressed in terms of the previously defined quantities as

$$E = \begin{cases} \cos^{-1} \dfrac{1}{e} \left[1 - \dfrac{R}{a}\right] \\ \sin^{-1} \dfrac{1}{e} \left[\dfrac{\underline{R} \cdot \underline{V}}{[\mu_\ell \, a]^{1/2}}\right] \end{cases}$$

The mean motion

$$n = \sqrt{\frac{\mu_\ell}{a^3}}$$

The true anomaly is given by

$$\cos \nu = \frac{\cos E - e}{1 - e \cos E}$$

$$\sin \nu = \frac{\sqrt{1 - e^2} \sin E}{1 - e \cos E}$$

Time of perifocal passage is given by

$$\tau = \frac{1}{n} [E - e \sin E]$$

The equations for the hyperbolic orbits are very similar. and will not be stated here.

f. For each time point, Kepler's equation

$$n\tau = E - e \sin E$$

is solved and the in-plane position and velocities are computed relative to a coordinate system with the ζ_1-axis pointing towards pericenter **according to the equations.**

$$\zeta_1(t_k^m) = a(t_o^m)\,[\cos E(t_k^m) - e(t_o^m)]$$

$$\zeta_2(t_k^m) = a(t_o^m)\,\sqrt{1 - e^2(t_o^m)}\,\sin E(t_k^m)$$

$$\dot{\zeta}_1(t_k^m) = a(t_o^m)\,n(t_o^m)\,[\frac{-\sin E(t_k^m)}{1 - e(t_o^m)\cos E(t_k^m)}]$$

$$\dot{\zeta}_2(t_k^m) = a(t_o^m)\,n(t_o^m)\,\sqrt{1 - e^2(t_o^m)}\,[\frac{\cos E(t_k^m)}{1 - e(t_o^m)\cos E(t_k^m)}]$$

For the case hyperbolic orbits, see page 3-106 of this part.

g. Since the guidance and navigation analysis is done in an equatorial coordinate system, the nominal position $\underline{R}^*(t_k^m)$ and velocity $\underline{V}^*(t_k^m)$ are obtained from the in-plane

$$\xi_1(t_k);\ \xi_2(t_k);\ \dot{\xi}_1(t_k);\ \dot{\xi}_2(t_k)$$

coordinates by the transformation $\mathcal{R}^*(t_o^m)$.

This completes the description of the equations used in the determination of the nominal orbits.

2.1.2 State Transition Matrices

The equation of motion during free-fall has the general form

$$\underline{\dot{x}} = \underline{f}(\underline{x})$$

and the linear perturbation equation is simply

$$\underline{\dot{x}} = F(t)\,\underline{x}$$

where

$$F(t) = \frac{\partial \underline{f}}{\partial \underline{X}} = \frac{\partial}{\partial \underline{X}}\left(\frac{\partial \underline{V}}{\partial \underline{X}}\right)$$

and V is the potential of the central body, i.e., $V = \mu/R$, $R = [X_1^2 + X_2^2 + X_3^2]^{1/2}$. The solution of the perturbation equation yields the so-called state transition matrix $\Phi(t, t_o)$ so that

$$\underline{x}(t) = \Phi(t, t_o)\,\underline{x}(t_o)$$

As was stated in Volume I, the state transition matrix can be solved for in closed form without it being necessary to evaluate the F(t) matrix. As in the nominal case, the basic calculations are performed here too in an in-plane coordinate system. In this system, the state transition for a specific conic section is computed from the eccentric anomaly and the eccentricity of the orbit.

To illustrate the computational procedure, let us suppose that we have only a single conic section and it is desired to obtain the state transition between two time points t_{k-1} and t_k. First, the state transition from the pericenter of the conic to the τ_{k-1} (corresponding to t_{k-1}) time is computed using the eccentricity e and corresponding eccentric anomaly E_{k-1}, i.e., $\phi(\tau_{k-1}, 0)$. Next $\phi(\tau_k, 0)$ is computed. From these two matrices, the $\phi(\tau_k, \tau_{k-1})$ matrix is computed as follows:

$$\phi(\tau_k, t_{k-1}) = \phi(\tau_k, 0)\,\phi^{-1}(\tau_{k-1}, 0)$$

where $\phi^{-1}(\tau_{k-1}, 0)$ is obtained through transposition of the elements of the ϕ, i.e., if

$$\phi(\tau_{k-1}, 0) = \begin{pmatrix} \phi_1 & \phi_2 \\ \phi_3 & \phi_4 \end{pmatrix}$$

then

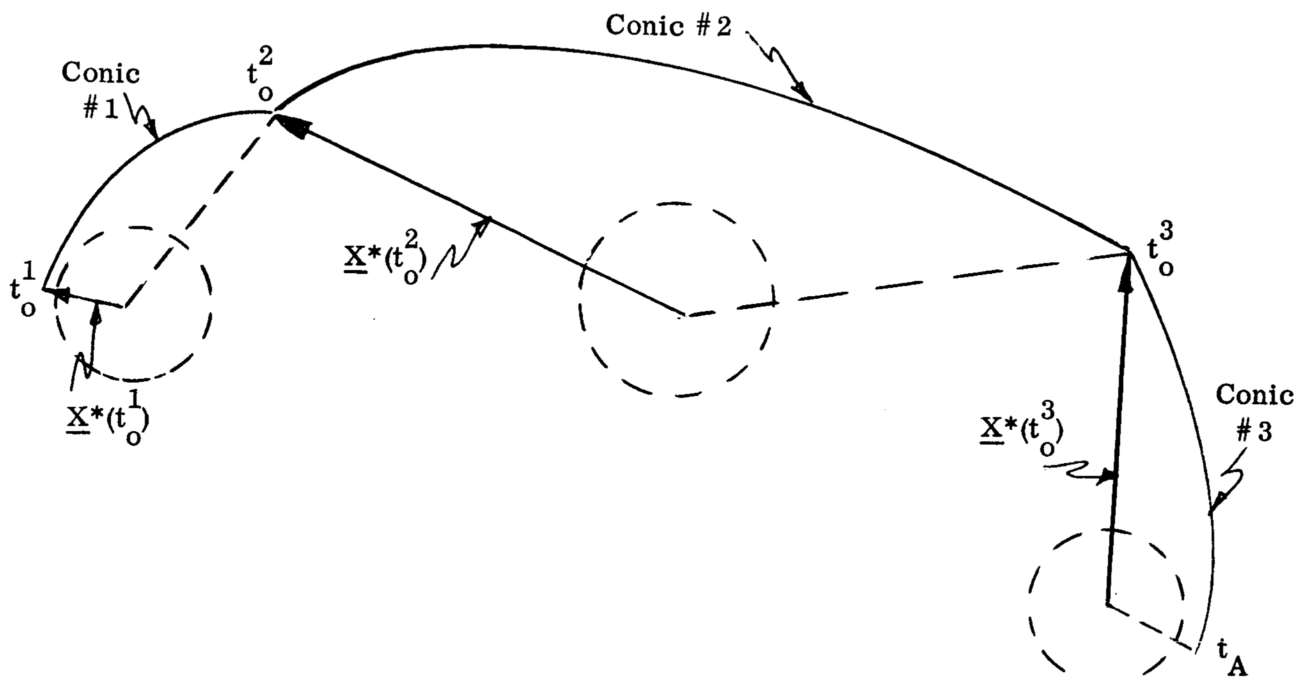
$$\phi^{-1}(\tau_{k-1}, 0) \overset{Df}{=} \phi(0, \tau_{k-1}) = \begin{pmatrix} \phi_4^T & -\phi_2^T \\ -\phi_3^T & \phi_1^T \end{pmatrix}$$

The state transition matrix in the basic non-rotating coordinate system is then given by

$$\Phi(t_k, t_{k-1}) = \mathcal{R}^*(t_o)\,\phi(\tau_k, \tau_{k-1})\,\mathcal{R}^{*T}(t_o)$$

where the $\mathcal{R}^*(t_o)$ is as defined in Paragraph 2.1.1.

In the program, it is also sometimes desirable to compute the total state transition accross several two body conic sections. This is simply accomplished by multiplying appropriately the individual state transition matrices. For example, suppose we have three conics sections, see figure below,



and it is desirable to obtain $\Phi(t_A, t_o')$. The overall state transition is computed thusly,

$$\Phi(t_A, t_o^1) = \Phi(t_A, t_o^3)\, \Phi(t_o^3, t_o^2)\, \Phi(t_o^2, t_o^1)$$

where

$$\Phi(t_o^3, t_o^2) = \mathcal{R}^*(t_o^2)\, \phi(\tau_o^3, 0)\, \phi^{-1}(\tau_o^2, 0)\, \mathcal{R}^{*T}(t_o^2)$$

The equations for the in-plane coordinate system state transition ϕ have the following structure.

Let

$$\Phi \overset{\mathrm{Df}}{=} \begin{bmatrix} \Phi_1 & \Phi_2 \\ \Phi_3 & \Phi_4 \end{bmatrix}$$

where the submatrices have the form

$$\Phi_1 = \begin{bmatrix} \Phi_{11} & \Phi_{12} & 0 \\ \Phi_{21} & \Phi_{22} & 0 \\ 0 & 0 & \Phi_{33} \end{bmatrix} \quad ; \quad \Phi_2 = \begin{bmatrix} \Phi_{14} & \Phi_{15} & 0 \\ \Phi_{24} & \Phi_{25} & 0 \\ 0 & 0 & \Phi_{36} \end{bmatrix}$$

$$\Phi_3 = \begin{bmatrix} \Phi_{41} & \Phi_{42} & 0 \\ \Phi_{51} & \Phi_{52} & 0 \\ 0 & 0 & \Phi_{63} \end{bmatrix} \quad ; \quad \Phi_4 = \begin{bmatrix} \Phi_{44} & \Phi_{45} & 0 \\ \Phi_{54} & \Phi_{55} & 0 \\ 0 & 0 & \Phi_{66} \end{bmatrix}$$

Let the eccentric anomaly at t_k be E_k and let the eccentricity and mean angular rate be e and n. Define S_k and C_k as

$$\sin E_k = S_k \quad ; \quad \cos E_k = C_k$$

Then, the elements of the submatrices are

$$\Phi_{11} = \frac{1}{(1-e)^2(1-e\,C_k)} \left[C_k^2(1+e-e^2) + C_k(2+e+2e^2-e^3) - 2 - 5e + 2e^2 + 3E_k S_k\right]$$

$$\Phi_{12} = \frac{\sqrt{1-e^2}}{(1-e)(1-e\,C_k)}\, S_k\,(1-C_k)$$

$$\Phi_{21} = \frac{\sqrt{1-e^2}}{(1-e)^2(1-e\,C_k)} \left[S_k C_k(1+e) + S_k(2-e) - 3E_k C_k\right]$$

$$\Phi_{22} = \frac{1}{(1-e)(1-e\ C_k)} \left[C_k^2 + C_k(-1-2e+e^2) + 1\right]$$

$$\Phi_{33} = \frac{(C_k - e)}{(1-e)}$$

$$\Phi_{14} = \frac{(1-e)}{n(1-e\ C_k)} S_k\left[-C_k(1+e) + 2\right]$$

$$\Phi_{15} = \frac{\sqrt{1-e^2}}{(1-e)n(1-e\ C_k)} \left[C_k^2(2-e) + 2C_k(1+e) - 4 - e + 3E_k S_k\right]$$

$$\Phi_{24} = \frac{\sqrt{1-e^2}}{n(1-e\ C_k)} \left[1 - C_k\right]^2$$

$$\Phi_{25} = \frac{1}{n(1-e\ C_k)} \left[S_k C_k(2+e+e^2) + 2S_k - 3(1+e)E_k C_k\right]$$

$$\Phi_{36} = \frac{S_k(1-e)}{n}$$

$$\Phi_{41} = \frac{n}{(1-e)^2(1-e\ C_k)^3} \left[S_k C_k^2(e+e^2-e^3) + S_k C_k(-2-5e+2e^2) + S_k(1+e+3e^2-e^3) + 3E_k(C_k-e)\right]$$

$$\Phi_{42} = \frac{n\sqrt{1-e^2}}{(1-e)(1-e\ C_k)^3} \left[e\ C_k^3 - 2C_k^2 + C_k + 1 - e\right]$$

$$\Phi_{51} = \frac{n\sqrt{1-e^2}}{(1-e)^2(1-e\ C_k)^3} \left[-C_k^3(e+e^2) + C_k^2(2+5e) - C_k(1+e) - 1 - 3e + e^2 + 3E_k S_k\right]$$

$$\Phi_{52} = \frac{n\ S_k}{(1-e)(1-e\ C_k)^3} \left[e\ C_k^2 - 2C_k + 1 + e-e^2\right]$$

$$\Phi_{63} = \frac{-n\ S_k}{(1-e)(1-e\ C_k)}$$

$$\Phi_{44} = \frac{1-e}{(1-e\ C_k)^3}\ [C_k^3(e+e^2) - 2C_k^2(1+e) + 2C_k + 1 - e]$$

$$\Phi_{45} = \frac{\sqrt{1-e^2}}{(1-e)(1-e\ C_k)^3}\ [S_k C_k^2(2e-e^2) - S_k C_k(4+e) + S_k(1+e)^2 + 3E_k(C_k-e)]$$

$$\Phi_{54} = \frac{S_k\sqrt{1-e^2}}{(1-e\ C_k)^3}\ [e\ C_k^2 - 2C_k + 2 - e]$$

$$\Phi_{55} = \frac{1}{(1-e\ C_k)^3}\ [-C_k^3(2e+e^2+e^3) + C_k^2(4+5e+5e^2)$$
$$-C_k(1+3e) - 2 - 3e - e^2 + 3(1+e)E_k S_k]$$

$$\Phi_{66} = \frac{(1-e)\ C_k}{(1-e\ C_k)}$$

Similar expressions can be obtained for hyperbolic orbits. For references, see Paragraph 3-110 of this part.

2.1.2 Actual Trajectory

The actual trajectory is computed according to the same equations and in the same manner as the nominal trajectory. No plant noise is assumed, so the only unknown relative to the actual trajectory is the initial state. In the program, the $\underline{X}_o$ can be specified by the engineer or can be obtained by establishing the perturbation $\underline{x}_o$ using a random noise generator and then forming

$$\underline{X}_o = \underline{X}_o^* + \underline{x}_o$$

For the details of how $\underline{X}_o$ is established, see page 3-49 of this part.

2.2 SENSORS

The Free-fall Performance Assessment Program has the capability of simulating several instruments. They are

(1) Space sextant

(2) Horizon sensor

(3) Planet tracker

(4) Ground-based radar — three radar stations can be included simultaneously.

2.2.1 Space Sextant

This instrument measures the angle between a star and the edge or center of a planet. The equations expressing these measurements are

$$\theta_{pb}(t_k) = \cos^{-1}\{\underline{s}(t_k) \cdot \underline{e}_{pb}(t_k)\} - \kappa_S \beta^S_{pb}(t_k)$$

where

$$\underline{e}_{pb}(t_k) = -\frac{\underline{R}_{pb}(t_k)}{R_{pb}(t_k)}$$

$$\beta^S_{pb}(t_k) = \sin^{-1}\frac{r_b}{R_{pb}(t_k)}$$

$$\underline{R}_{pb}(t_k) = \underline{R}(t_k) - \underline{R}_{bc}(t_k) \qquad \underline{R}_{bb} = 0$$

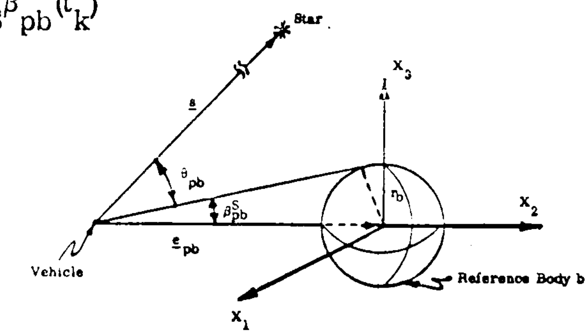


Figure 1. Space Sextant Measurement Geometry

r_b = radius of reference body $\qquad b \stackrel{Df}{=}$ BODY

$\underline{R}_{bc}(t_k)$ = position of reference body b with respect to central body c

and the unit vector from the vehicle to the star is

$$s_j(t_k) = \begin{bmatrix} s_{j1} \\ s_{j2} \\ s_{j3} \end{bmatrix} = \begin{bmatrix} \cos\alpha_j & \cos\delta_j \\ \sin\alpha_j & \cos\delta_j \\ \sin\delta_j & \end{bmatrix}$$

The α_j and δ_j define the right ascension and declination of the j^{th} star. By setting $\kappa_S = 0$ in the input, the center of the planet is used in the measurements.

The observation matrix $H(t_k)$ for this device has dimension (1×6). Since the angle θ_{pb} does not depend upon the velocity, the partial derivatives involving the velocity are identically zero. The remaining elements are:

$$h_{11}(t_k) = \frac{\partial \theta_{pb}}{\partial X_1} \quad ; \quad h_{12}(t_k) = \frac{\partial \theta_{pb}}{\partial X_2} \quad ; \quad h_{13}(t_k) = \frac{\partial \theta_{pb}}{\partial X_3}$$

Further definitions of the partials are given in Volume I and on page 3-163 of this part.

The star that is used at a particular sampling time (t_k) can be specified arbitrarily through input or it can be selected to minimize the mean-square error in the estimate of the perturbation in the terminal constraints. In the "optimal" star selection option, the quantity that is minimized at each sampling time (t_k) is the trace of

$$T_E(t_A) = T(t_A)P(t_A)T^T(t_A)$$

with

$$P(t_A) = \Phi(t_A, t_k)[I - P(t_k)H^T(t_k)\{H(t_k)P(t_k)H^T(t_k) + R(t_k)\}^{-1}H(t_k)]P(t_k)\Phi^T(t_A, t_k)$$

$T(t_A)$ = transformation matrix from state to constraints at terminal time t_A

P = covariance matrix of the error in the best estimate

Φ = state transition matrix

R = covariance matrix of the instrument measurement noise.

Because in the space sextant case there is only one measurement θ_{pb}, the quantity $\{HPH^T + R\}^{-1}$ above is a scalar. Furthermore, since the only measurement dependent matrix is H, the minimization of the trace of $T_E(t_A)$ is equivalent to maximizing the trace of

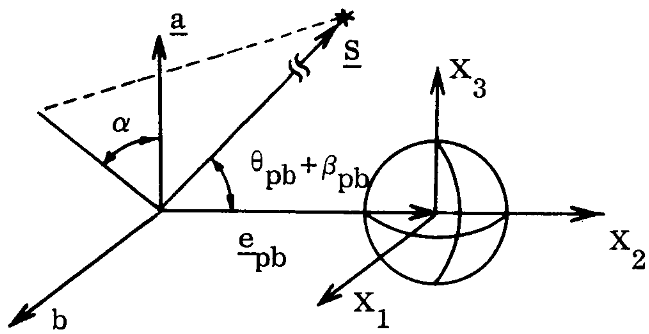
$$\frac{T(t_A)\Phi(t_A, t_k)P(t_k)H^T(t_k)H(t_k)P(t_k)\Phi^T(t_A, t_k)T^T(t_A)}{H(t_k)P(t_k)H^T(t_k) + R(t_k)}$$

$$\underset{=}{\mathrm{Df}} \quad \frac{H(t_k)P(t_k)\Phi^T(t_A, t_k)T^T(t_A)T(t_A)\Phi(t_A, t_k)P(t_k)H^T(t_k)}{H(t_k)P(t_k)H^T(t_k) + R(t_k)}$$

We shall now show that the above maximization problem reduces to finding the maxima of a function of a single variable, an angle. From the measurement equation, one notes that the non-trivial portion of the observation matrix is given by

$$\underline{h} = \frac{1}{R_{pb} \sin (\theta_{pb} + \varkappa_S \beta_{pb})} [\underline{S} - (\frac{\cos \theta_{pb}}{\cos \varkappa_S \beta_{pb}}) \underline{e}_{pb}]$$

Let us now introduce a new orthogonal coordinate system through the unit vectors $\underline{e}_{pb}$, $\underline{a}$, $\underline{b}$ and choose $\underline{a} = (0, a_1, a_2)^T$, see figure.



In this coordinate system, the $\underline{h}$ vector becomes

$$\underline{h} = \frac{1}{R_{pb}} [1 \;\; \cos \alpha \;\; \sin \alpha] \; E$$

with

$$E = \begin{bmatrix} - t_\beta e_1 & - t_\beta e_2 & - t_\beta e_3 \\ 0 & e_3/c & - e_2/c \\ - c & e_1 e_2/c & e_1 e_3/c \end{bmatrix}$$

$$\underline{e} = (e_1, e_2, e_3)^T$$

$$t_\beta = \tan \varkappa_S \beta_{pb}$$

$$c = (e_2^2 + e_3^2)^{1/2}$$

With the above property for the observation matrix, the trace to be maximized reduces to a function of a single variable α, i.e.,

$$Q(\alpha) = \frac{(1, \cos\alpha, \sin\alpha)\, A^* \,(1, \cos\alpha, \sin\alpha)^T}{(1, \cos\alpha, \sin\alpha)\,(B^* + R)\,(1, \cos\alpha, \sin\alpha)^T}$$

To obtain the "optimal" star direction, it remains to find $\alpha_{optimal}$ which gives a maximum for $Q(\alpha)$. For further details as well as a definition of A^* and B^*, see page 3-167 through 3-171 of this part.

2.2.2 Horizon Sensor

This instrument provides information about three angles

Local elevation angle:

$$\alpha = -\sin^{-1} \frac{X_3(t_k)}{|\underline{R}\ (t_k)|}$$

Local azimuth angle:

$$\delta = \sin^{-1} \frac{X_2(t_k)}{[X_1^2(t_k) + X_2^2(t_k)]^{1/2}}$$

Half-subtended angle:

$$\beta^H = \sin^{-1} \frac{r_\ell}{|\underline{R}\ (t_k)|}$$

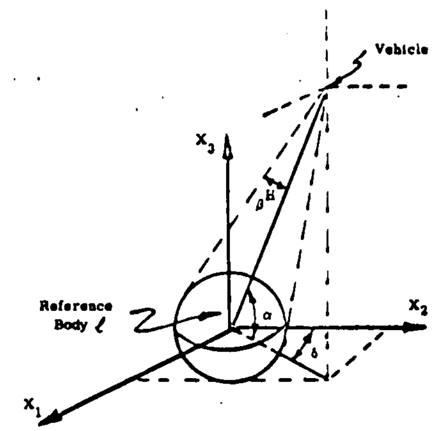


Figure 2. Horizon Sensor Geometry of Angular Measurements

The range of applicability of the horizon sensor is governed in the program by the subtended angle. If $\beta^H_{min} < \beta^H \leq \beta^H_{max}$, the horizon sensor information is used, otherwise it is ignored. For further details, see pages 3-161 and 4-26 of this part.

Since the horizon sensor provides information about three angles, the observation matrix $H(t_k)$ has the dimension (3 x 6). The partial derivatives with respect to velocity are zero; therefore, the non-trivial portion of this matrix has dimension (3 x 3). For a definition of this matrix, see page 3-163 of this part.

2.2.3 Planet Tracker

The horizon sensor equations can be re-interpreted as equations for a planet tracker if the information about the subtended angle is ignored. For more details about this option, see page 4-26 of this part.

2.2.4 Ground Tracking

Range, range rate, and angular information is provided according to the following equations.

The range and range rate vector equations are given by

$$\underline{\rho}_i(t_k) = \underline{R}(t_k) - \underline{r}_{iT}(t_k) + \underline{R}_{\ell o}(T_k)$$

$$\dot{\underline{\rho}}_i(t_k) = \underline{V}(t_k) - \dot{\underline{r}}_{iT}(t_k) + \dot{\underline{R}}_{\ell o}(T_k)$$

$$i = 1, 2, 3$$

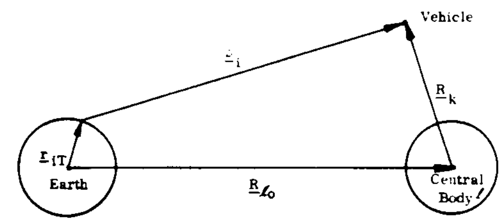


Figure 3. Range Vector Relationships

where

$$\underline{R}(t_k) \overset{Df}{=} [X_1(t_k), X_2(t_k), X_3(t_k)]^T \text{ and } \underline{V}(t_k) \overset{Df}{=} [X_4(t_k), X_5(t_k), X_6(t_k)]^T$$

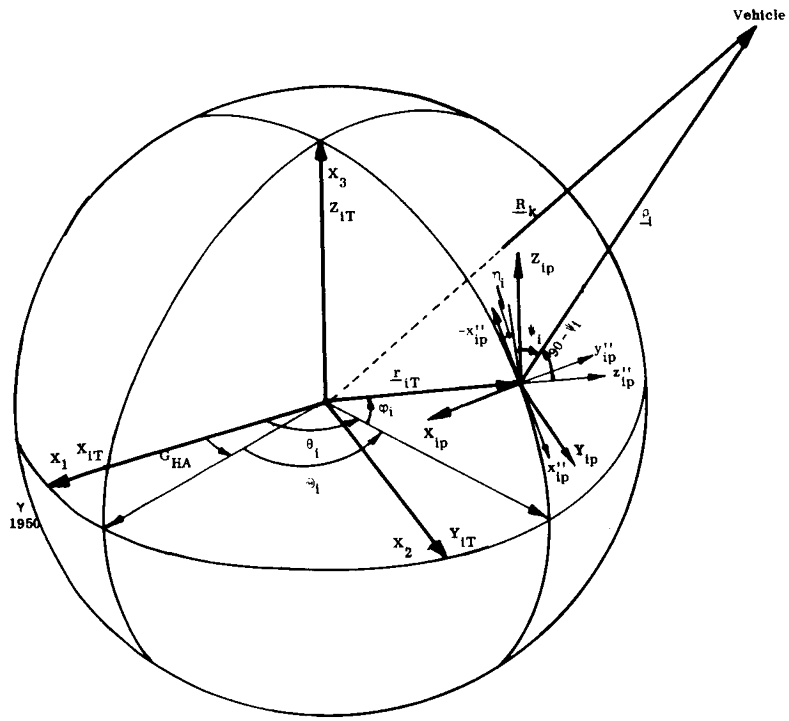
are obtained by evaluating the two body equations in the in-plane coordinates and transforming to the non-rotating coordinates at each t_k. $\underline{R}_{\ell o}(T_k)$ and $\dot{\underline{R}}_{\ell o}(T_k)$ are the position and velocity of the ℓ^{th} planet with respect to the Earth (in equatorial coordinates) obtained from an ephemeris at each epoch time $T_k \leftrightarrow t_k$.

$$\underline{r}_{iT}(t_k) \overset{Df}{=} \begin{bmatrix} X_{iT}(t_k) \\ Y_{iT}(t_k) \\ Z_{iT}(t_k) \end{bmatrix} = \begin{bmatrix} r_{iT} \cos \varphi_i \cos (\theta_i + \omega t_k) \\ r_{iT} \cos \varphi_i \sin (\theta_i + \omega t_k) \\ r_{iT} \sin \varphi_i \end{bmatrix}$$

and

$$\dot{\underline{r}}_{iT}(t_k) \overset{Df}{=} \begin{bmatrix} -\omega Y_{iT}(t_k) \\ \omega X_{iT}(t_k) \\ 0 \end{bmatrix}$$

are the tracker position and velocity in an Earth centered system, see Figure 4.



$(r_{iT}, \theta_i, \varphi_i)$	Earth fixed tracker coordinate system
$(x''_{ip}, y''_{ip}, z''_{ip})$	Earth fixed azimuth-elevation tracker coordinate system
(X_1, X_2, X_3)	Non-rotating right-handed planet centered equatorial coordinate system with X_1 along the vernal equinox of 1950
(X_{iT}, Y_{iT}, Z_{iT})	Non-rotating right-handed Earth centered tracker coordinate system parallel to the (X_1, X_2, X_3) system
(X_{ip}, Y_{ip}, Z_{ip})	Non-rotating right-handed tracker centered coordinate system parallel to the (X_1, X_2, X_3) system

Figure 4. Definition of Tracker Coordinate Systems

Defining the vehicle, with respect to tracker location, position vector as

$$\underline{\rho}_i(t_k) \overset{Df}{=} \begin{bmatrix} X_{ip}(t_k) \\ Y_{ip}(t_k) \\ Z_{ip}(t_k) \end{bmatrix}$$

Then the range and range rate from the trackers are given by

$$\rho_i = [X_{ip}^2 + Y_{ip}^2 + Z_{ip}^2]^{1/2}$$

$$\dot{\rho}_i = \frac{\underline{\rho}_i^T \cdot \dot{\underline{\rho}}_i}{\rho_i}$$

Elevation and azimuth angle of probe with respect to the tracker

$$\psi_i = \sin^{-1}\left[\frac{\underline{r}_{iT}^T \underline{\rho}_i}{r_{iT}\,\rho_i}\right] \qquad -90^\circ < \psi_i < 90^\circ$$

$$\eta_i = \begin{cases} \cos^{-1}\left[\dfrac{-x''_{ip}}{\rho_i \cos\psi_i}\right] \\ \sin^{-1}\left[\dfrac{y''_{ip}}{\rho_i \cos\psi_i}\right] \end{cases} \qquad 0 \le \eta_i \le 360^\circ$$

where

$$\begin{bmatrix} x''_{ip} \\ y''_{ip} \end{bmatrix} = \begin{bmatrix} \sin\varphi_i \cos(\theta_i + \omega t_k) & \sin\varphi_i \sin(\theta_i + \omega t_k) & -\cos\varphi_i \\ -\sin(\theta_i + \omega t_k) & \cos(\theta_i + \omega t_k) & 0 \end{bmatrix} \begin{bmatrix} X_{ip} \\ Y_{ip} \\ Z_{ip} \end{bmatrix}$$

In the program, each tracker elevation angle ψ_i is tested against input quantities ψ_{io} to see if the vehicle is visible from the tracker. If $\psi_{io} \le \psi_i \le 1.356$ radians, then the measurements from the tracker are processed. In addition to the elevation test, two additional simple tests are performed in the program to establish whether the Sun is in the vehicle-Earth line of sight or if the vehicle is behind the central body planet. For details of these two tests, see pages 3-148 and 4-22, 4-23.

The ground trackers measure the range ρ_i and range rate $\dot{\rho}_i$ of the vehicle and its elevation ψ_i and azimuth η_i angles. This leads to a basic observation matrix that is (4x6).

In addition to these partial deviatives, tracker location errors and bias errors are included. Each tracker location is described by the vector components X_{iT}, Y_{iT}, Z_{iT}. In order to include these constant errors, an observation matrix relating errors in the measured quantities ρ_i, $\dot{\rho}_i$, ψ_i, η_i to the tracker errors must be established. The bias errors are assumed to enter additively so the observation matrix for the tracker errors has the dimension (4 x 7). For further details and a definition of the partial derivatives in the observation matrices for ground tracking, see pages 3-158 through 3-160 of this part.

The facility has been included in the program to consider bias errors in the measurement data. When this option is utilized, the biases are considered as additional state variables.

The bias errors are random quantities and are described as having a mean value of zero and covariance matrix B_o. In the simulation, the B_o is used in conjunction with a noise generator to obtain numerical values for the bias errors. The measurement data also are corrupted by random noise. These random errors are defined as having zero mean. At each sampling time, the covariance matrix of the noise is assumed to be known and is denoted by R_k. Random noise is obtained from a noise generator using these statistics.

2.3 NAVIGATION

The Kalman filter is used to estimate the state perturbations from the measurement data. The plant model is simpler than the general form described in Volume I, so the filter equations can also be simplified. In the Free-fall Performance Assessment Program, the general form of the Kalman filter is given by

$$\hat{\underline{x}}_k = \hat{\underline{x}}'_k + K_k (\underline{z}_k - H_k \hat{\underline{x}}'_k)$$

where

$$\hat{\underline{x}}'_k = \Phi_{k,k-1} \hat{\underline{x}}_{k-1}$$

$$K_k = P'_k H_k^T [H_k P'_k H_k^T + R_k]^{-1}$$

$$P'_k = \Phi_{k,k-1} P_{k-1} \Phi^T_{k,k-1} + Q_{k-1}$$

$$P_k = P'_k - K_k H_k P'_k$$

Although the plant does not explicitly contain any noise, the matrix Q_{k-1} has been included in the equations. This feature is discussed below.

The only difference between these equations and those in Volume I is the absence of the known plant term $\underline{f}_{k-1}$. There are several aspects relating to these general equations that are not immediately apparent. First, the state vector must be defined. The error covariance matrix P_k is of the same order as the dimension of the state. Theoretically, the P_k is a non-negative definite, symmetric matrix. Because of computational inaccuracies, it can happen that the P_k loses both of these properties. Techniques have been devised that delay or prevent this from occurring. The inaccuracies result from the finite word-length of a digital computer. Capability has been included that allows one to simulate the effects of a reduced word-length. These topics will be discussed briefly in the subsequent paragraphs.

2.3.1 The State Vector

The state vector is at least six-dimensional and always contains the position and velocity as components. In this program, the bias errors of the electromagnetic sensors provide the other possible components of the state vectors. The most general state vector occurs when every instrument is included and has the following form.

$$\left.\begin{matrix} X_1 \\ X_2 \\ X_3 \end{matrix}\right\} \text{ position of the spacecraft}$$

$$\left.\begin{matrix} X_4 \\ X_5 \\ X_6 \end{matrix}\right\} \text{ velocity of the spacecraft}$$

$$\left.\begin{matrix} X_7 \\ X_8 \\ X_9 \end{matrix}\right\} \text{ errors in location of Tracker No. 1}$$

$$\left.\begin{matrix} X_{10} \\ X_{11} \\ X_{12} \\ X_{13} \end{matrix}\right\} \text{ bias errors in measurements from Tracker No. 1}$$

$$\left.\begin{matrix} X_{14} \\ X_{15} \\ X_{16} \end{matrix}\right\} \text{ errors in location of Tracker No. 2}$$

$$\left.\begin{array}{l} X_{17} \\ X_{18} \\ X_{19} \\ X_{20} \end{array}\right\} \text{ bias errors in measurements from Tracker No. 2}$$

$$\left.\begin{array}{l} X_{21} \\ X_{22} \\ X_{23} \end{array}\right\} \text{ errors in location of Tracker No. 3}$$

$$\left.\begin{array}{l} X_{24} \\ X_{25} \\ X_{26} \\ X_{27} \end{array}\right\} \text{ bias errors in measurements from Tracker No. 3}$$

$$\left.\begin{array}{l} X_{28} \\ X_{29} \\ X_{30} \end{array}\right\} \text{ bias errors in horizon sensor measurements}$$

$$X_{31} \quad \text{bias error in space sextant measurement.}$$

Thus, the maximum dimension of the state vector in this program is thirty-one.

2.3.2 Orbit Rectification

The estimation procedure can be reliable only when the linear perturbation theory is valid. In an effort to improve the linear approximation, it is possible to exchange nominal trajectories at any intermediate time by re-initializing the conic. This rectification is accomplished when two tests are satisfied.

$$\sum_{i=1}^{3} p_{ii} < k_P$$

$$\sum_{i=4}^{6} p_{ii} < k_V$$

The p_{ii} are diagonal elements of the $P(t_k)$ matrix, so they represent the variance of the error in the estimate of the i^{th} component of the state vector. Recall that the first three components define position and the next three define velocity. The k_P and k_V are arbitrary constants.

2.3.3 Square-root Kalman Filter

Since P_k is non-negative definite and symmetric, it is always possible to write it in terms of a matrix factor Π_k

$$P_k = \Pi_k \Pi_k^T$$

When there is no plant noise and when each measurement is processed individually, it is possible to obtain recursion relations for the Π_k. Then, one can base the computations on the Π_k. Since P_k is then the product of a matrix and its transpose, one would expect that P_k would remain symmetric and non-negative definite. This formulation of the Kalman filter is included in the program and is described in more detail in Volume I and page 3-134 of this part.

2.3.4 Compensation for Computational Errors

The matrix Q_{k-1} that appears in the filter equations generally represents the covariance of the plant noise. In this program, no plant noise exists, so an additional explanation regarding the presence of Q_{k-1} is appropriate.

It can be demonstrated that the presence of plant noise will, in general, prevent the error covariance matrix from tending to vanish. It is this tendency that causes many of the problems that arise with the P_k (i.e., the loss of non-negative definiteness). One can consider the computational errors to have the general effect of a random noise on the system. Then, the Q_{k-1} can be used to compensate for the errors and to prevent the P_k from vanishing. Furthermore, the Q_{k-1} can be considered to compensate for errors in the linear approximation.

2.3.5 Variable Computer Word-length

Provision has been made to examine some of the effects of a reduced computer word-length. This is accomplished by setting a specified number of bits of the elements of P_k, K_k, and $\hat{\underline{x}}_k$ equal to zero at the end of every computational cycle.

2.4 GUIDANCE

It is assumed that the trajectory corrections that are required for this type of mission will be small and that they can be adequately represented by an instantaneous change in the velocity of the vehicle. Based on this assumption, the guidance policy involves two primary considerations

(1) The time of the velocity correction must be established. Needless to say, velocity correction will not be made at every sampling time. It is, in fact, desirable to introduce as few corrections as is necessary to accomplish the mission objectives.

(2) The velocity correction itself must be determined according to some criterion.

The policy used in the program is presented below. The guidance law is then discussed in terms of the general statements of Section 2.

2.4.1 Determination of the Desired Velocity Correction

The guidance laws are determined by the terminal conditions required for the mission. At each correction time, the velocity that is required to completely null the error in specified terminal constraints is computed using the linear perturbation theory and the current best estimate. In general, the correction is determined according to

$$\Delta\hat{\underline{V}}_k = \Lambda_k \hat{\underline{x}}_k$$

where $\hat{\underline{x}}_k$ represents the best estimate of the basic six-dimensional state. The guidance matrix Λ_k is established in a deterministic manner to accomplish the objective. Five different control laws have been considered and are listed below. They are classified by the nature of the terminal constraints.

In some cases, the time of arrival at the target is allowed to be different than the nominal. The change is computed from

$$\delta T = C_T \hat{\underline{x}}_k$$

1. Fixed position - fixed time of arrival

   The spacecraft is required to reach a specified position at a given time.

2. Fixed position - variable time of arrival

   The spacecraft is required to arrive at a specified position; the time at which it accomplishes this objective is not restrained. In this case, the magnitude of the velocity correction is minimized through the choice of the time of arrival.

3. R, γ, θ - fixed time of arrival

   The spacecraft is required to achieve a desired radial distance, flight path angle, and velocity direction at the nominal time of arrival.

4. R, γ, θ - variable time of arrival

   This control law is similar to the preceding one except the time of arrival is not specified. It is chosen to minimize the magnitude of the velocity correction.

5. $R, \gamma, \underline{u}$ - variable time of arrival

In addition to the radial distance and flight path angle, the position and velocity of the vehicle is required to lie in a desired plane. The time of arrival is used to satisfy the fourth constraint that is introduced by the constraint on $\underline{u}$.

The specified form of the Λ and C_T are given in the Program Definition.

2.4.2 Timing of the Velocity Corrections

Several techniques have been suggested in the literature. In this program, the variance ratio criteria suggested by Battin is used. Using this approach, the time of the correction is based upon statistics relating to the velocity correction.

The covariance of the estimated correction is

$$\begin{aligned} V(t_k) &= E[\Delta\hat{\underline{V}}(t_k)\,\Delta\hat{\underline{V}}^T(t_k)] \\ &= \Lambda(t_k)\,[M_k - P_k]\,\Lambda^T(t_k) \end{aligned}$$

where

$$M_k = \Phi_{k,k-1} M_{k-1} \Phi^T_{k,k-1}$$

$$P_k = \Phi_{k,k-1} P_{k-1} \Phi^T_{k,k-1}$$

The trace of this matrix is defined as $\overline{\Delta V^2(t_k)}$.

The error in the estimate of the velocity correction is characterized statistically by its covariance matrix

$$\begin{aligned} D(t_k) &= E[(\Delta\hat{\underline{V}}(t_k) - \Delta\hat{\underline{V}}_e(t_k))\,(\Delta\hat{\underline{V}}(t_k) - \Delta\hat{\underline{V}}_e(t_k))^T] \\ &= \Lambda(t_k)\,P(t_k)\,\Lambda^T(t_k) \end{aligned}$$

where $\Delta\underline{V}_e(t_k)$ would be the velocity correction if the state were known perfectly. The trace of this matrix is defined as $\overline{d^2(t_k)}$.

A velocity correction is commanded according to the variance ratio criterion below. This test is accomplished by comparing the quantity

$$R_V = \frac{\overline{d^2(t_k)}}{\Delta V^2(t_k)}$$

with some arbitrary constant R_V min. When $R_V < R_V$ min, a correction is made.

The velocity correction is actually implemented in the program after three additional tests are passed. They are: the terminal time test $t_k \leq \kappa\, t_A$, the terminal constraint test $\kappa_i \geq \sqrt{T_{E_{ii}}}$, and the estimated velocity magnitude test $|\Delta\hat{\underline{V}}| \geq V_{min}$. For more details, see p. 3-130, 3-143, and 4-35 of this part.

2.4.3 Implementation of the Correction

The control system will, in general, not perform guidance corrections exactly. The errors in the control system and non-nominal engine performance are assumed to combine to result in actual velocity corrections whose magnitude are proportional to the estimate.

$$|\Delta\underline{V}(t_k)| = (1 + \eta)\, |\Delta\hat{\underline{V}}(t_k)|$$

The η is a Gaussian random variable with mean zero and variance $\overline{\eta^2}$. It is selected from a Gaussian number generator. The angle between the actual and estimated corrections if δ and is a Gaussian random variable with mean zero and variance $\overline{\delta^2}$.

The difference between the estimated and actual velocity corrections is characterized statistically by the covariance matrix

$$\begin{aligned} N(t_k) &= E[(\Delta\hat{\underline{V}}(t_k) - \Delta\underline{V}(t_k))\,(\Delta\hat{\underline{V}}(t_k) - \Delta\underline{V}(t_k))^T] \\ &= \overline{\eta^2}\, V(t_k) + \frac{\overline{\gamma^2}}{2}\,[v(t_k)\, I - V(t_k)] \end{aligned}$$

where

$$v(t_k) = \sum_{i=1}^{6} \sum_{j=1}^{6} \beta_{ij}\,[m_{ij}(t_k) - p_{ij}(t_k)]$$

and the m_{ij} and p_{ij} are elements of M_k and P_k. The β_{ij} are elements of the matrix $\Lambda^T(t_k)\, \Lambda(t_k)$.

At velocity correction times, the navigation statistics and state vector estimate must be modified to include the effects of the correction.

$$P_k = P'_k + J\, N_k\, J^T$$

$$M_k = [I + J\,\Lambda(t_k)]\,[M(t_k) - P(t_k)]\,[I + J\,\Lambda(t_k)]^T + P(t_k)$$

$$\hat{\underline{x}}_k = \Phi_{k,k-1}\,\hat{\underline{x}}_{k-1} + J\,\Delta\hat{\underline{V}}(t_k)$$

and

$$J = \begin{pmatrix} 0 \\ I \end{pmatrix}$$

The 0 and I represent 3 x 3 zero and identity matrices, respectively.

It should be noted that when a velocity correction occurs, no measurement data is processed. Thus, the $\hat{\underline{x}}_k$ and P_k given above provide the best available navigational data.

2.4.4 Relation to the General Guidance Problem

The guidance policy described above does not bear a close resemblance to the problem described in Section 2. It is interesting to note that the same guidance laws are obtained by formulating the velocity correction $\Delta\underline{V}$ as the control variables, assuming that only one correction will be made, and by then minimizing the expected value of the square of the terminal constraint errors. The same guidance laws are obtained. The requirement that the derivation assume that only one correction will occur illustrates the suboptimal character of this common guidance policy. Policies of this general character (i.e., policies which ignore the fact that additional data and corrections are to be available) are referred to as open-loop feedback control. One might expect that a closed-loop policy in which the total number of corrections is specified and thus is considered in the determination of each $\Delta\underline{V}_k$ would be superior. However, the present policy is used because of its common application in other space guidance studies.

3.0 COMPUTER PROGRAM DESCRIPTION

3.1 INTRODUCTORY AND EXPLANATORY REMARKS

In this document a digital computer program that can be used to study the guidance and navigation requirements for interplanetary space flights is described in very great detail. The bulk of the presentation is composed of flow charts and supporting equations. No attempt is made to justify the equations or to discuss their physical meaning except through definitions of symbols and nomenclature. For theoretical background, the reader is directed to References 1 through 8.

Flow charts provide the basic framework around which the remainder of the discussion is constructed. These diagrams serve to indicate the logical flow connecting different functional blocks. They do not describe literally the operation within the computer program itself because many of the programming details are of little interest to most engineers. It should be noticed, however, that the functional blocks have been assigned numbers and that these numbers will be used in future discussions of changes and modifications with programming personnel. We emphasize this aspect in order to point out the close relationship that exists between the program as described here and as it actually exists. Discussion of the programming details in terms of the actual program assembly shall be deferred to a later document.

The flow charts have been arranged and drawn according to a hierarchical structure. The "highest" level, designated as Level I, depicts the over-all structure of the program. Each block appearing in this chart is described by another flow chart. These charts are designated as Level II. This policy is repeated for each block in every level until no further logic remains to be described. In almost all cases, three levels of flow charts suffice to accomplish this objective. The final set of flow charts at the lowest level are supplemented by the detailed equations which are used in the program.

Paragraph 3.1.1 contains a further discussion and definition of the criteria used to establish the different flow chart levels. The symbols used in the flow charts are defined in Paragraph 3.1.2. The symbols and nomenclature that are fundamental to the discussion and equations are defined in Paragraph 3.1.3.

The Level I flow chart and its discussion constitute Paragraph 3.2. The remaining sections contain the charts and equations describing the functional blocks shown in this diagram. The functions that are not part of the basic computational cycle, INPUT GENERAL INITIALIZATION, and OUTPUT, are described in Paragraph 3.3. Paragraph 3.4 describes the computational blocks which are referred to as the TWO-BODY INTERPLANETARY NOMINAL, EXPLICIT STATE TRANSITION, TWO-BODY ACTUAL, GUIDANCE, ELECTROMAGNETIC SENSORS AND NAVIGATION blocks.

3.1.1 Schema for Flow Chart Presentation

As has already been stated, the flow charts are arranged according to "levels." In the resulting hierarchy, the Level I flow chart provides the most general description since it depicts the over-all program. Each functional block is further described by lower level flow charts. These charts indicate the logical flow within the block and describe the input and output requirements of the block. The equations used to obtain the desired outputs are presented as a supplement to the lowest level flow chart. The number of levels that are required depends upon the logical complexity of the functional block. In most cases three levels are required.

LEVEL I: This flow chart is designed to provide a very general description of the entire program. The titles assigned to the functional blocks are intended to be suggestive of the nature of the role to be performed within the block. Those functions that are to be performed in the basic computational cycle are designated by Roman numerals. Roman letters are used for functions that occur only once or play a passive role.

In a less complex program the input and output quantities required by the program could be described on this flow chart. However, this approach proved to be impracticable for this program so these requirements are described in the appropriately named functional blocks.

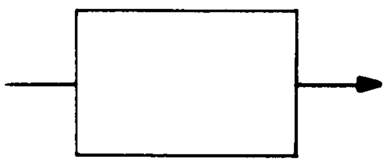
To indicate the basic logical decisions that can regulate and alter the flow between functional blocks, decision blocks are indicated. These decisions represent in a general manner the types of decisions that are required. The actual decision logic is described in the Level II flow charts of the functional blocks immediately preceding the decision block.

LEVEL II: The Level II flow charts provide the first concrete description of the program. Only the most important logical flow within each functional block is indicated on these diagrams. The quantities that are required for all logical and computational operations within this block are stated on this chart. These quantities are differentiated as being either INPUT (i.e., values provided initially by the engineer) or COMPUTED (i.e., values determimed in other portions of the program). The quantities that are required in other parts of the program, either for print-out or for computations, are also indicated on this flow chart. The functional blocks that appear on these diagrams are denoted by two symbols (e.g., II.1 when discussing the "first" block in the Level II flow chart of functional block II) and a name. The names have been selected to provide some insight into the nature of the block.

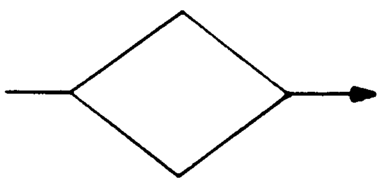
LEVEL III (and below): These diagrams provide additional details of the logical flow within the functional blocks depicted at Level II. In this program definition, Level III provide the description of the most intimate logical details in almost every case so no purpose was served by proceeding to lower levels. These flow diagrams are augmented by the equations programmed into the computer. The input and output requirements of these blocks are stated on the diagrams. All of these quantities are summarized on the Level II flow chart.

3.1.2 Definition of Flow Chart Symbols

The following symbols represent the only ones that are used in the flow charts presented below.

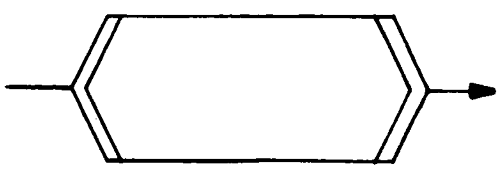


Set of operations that is to be described further by additional flow charts or by equations

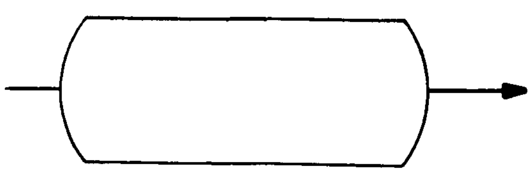


Logical Decision

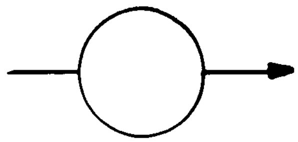
Operations that are predefined (i.e., in some other document and/or other parts of the program)



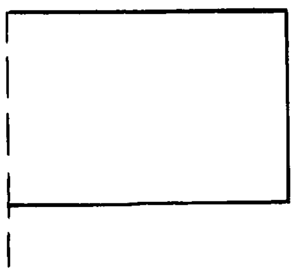
Operations are completely defined by the statements contained within the box



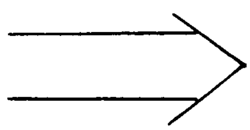
Connector used on Level II Flow Charts to indicate entry source and exit destination



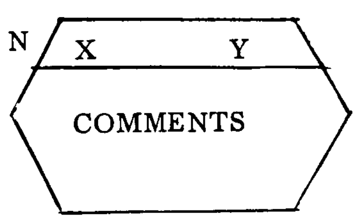
Connector used on Level III flow charts



Summary of all quantities required in computations of flow chart on which this symbol appears or, alternatively, summary of all quantities computed in this flow chart which are required in other operations



This broad arrow appears on Level I and Level II flow charts. It is used to indicate information flow from one block to another. The more important information is stated within the arrow. This symbol has been introduced to emphasize that many quantities are transmitted between the functional blocks in the higher level charts.



Indicates a call to a closed subroutine;
where N = FORTRAN statement number
X = Subroutine name
Y = Block reference to paragraph 3.0.

This symbol is used in paragraph 6.0.

3.1.3 Definition of Nomenclature

3.1.3.1 Notation

Symbol	Description
$\underline{(\ \ )}$	( ) is a vector
$(\ \ )^*$	The nominal value of ( )
$(\ \ )^T$	The matrix transpose of ( )
$(\ \ )^{-1}$	The matrix inverse of ( )
$(\ \ )_{ij}$	The ij element of the matrix ( )
$(\ \hat{}\ )$	Best linear estimate of ( )
$\overline{(\ \ )^2}$	Variance of ( )
$(\ \ )_o$	Initial value of ( )
$(\ \ )_i$	i^{th} sensor value of ( )
$(\ \ )_A$	The augmented value of ( )
$(\ \ )_f$	The final value of ( )

3.1.3.2 Definition of Flags

Symbol	Description
ACTFG	Determines source of initial conditions for actual trajectory. NEEDED ONLY WHEN PRFG = 1. $\text{ACTFG} = \begin{cases} 0, & \text{input} \\ 1, & \text{generated from initial covariance } M(t_o) \end{cases}$
BODY	Reference body flag; defines body to be used as a reference in computing the horizon sensor angles and the space sextant angle. This quantity is NEEDED ONLY WHEN HSFG $\neq 0$ and/or SSFG $\neq 0$. $\text{BODY} = \begin{cases} 0 & \text{Earth} \\ 1 & \text{Moon} \\ 2 & \text{Sun} \\ 3 & \text{Venus} \\ 4 & \text{Mars} \\ 5 & \text{Saturn} \\ 6 & \text{Jupiter} \end{cases}$

3.1.3.2 Definition of Flags (con't)

Symbol	Description
BSFG	Bias error flag; indicates that constant random errors (i.e., bias errors) are included in the model of the observation processes. State vector is augmented when non-zero. $$\text{BSFG} = \begin{cases} 0, & \text{no bias errors} \\ 1, & \text{bias errors} \end{cases}$$
CNTRFG	Central-body flag; defines the central body used for the two-body trajectory computations. $$\text{CNTRFG} = \begin{cases} 0, & \text{Earth} \\ 1, & \text{Moon} \\ 2, & \text{Sun} \\ 3, & \text{Venus} \\ 4, & \text{Mars} \\ 5, & \text{Saturn} \\ 6, & \text{Jupiter} \end{cases}$$
CORDFG	Coordinate conversion flag; converts certain output parameters to a rotating cartesian tangent, normal, radial coordinate system. $$\text{CORDFG} = \begin{cases} 0, & \text{no conversion} \\ 1, & \text{conversion} \end{cases}$$
CORRFG	Correction flag; normally set equal to zero. When non-zero, a velocity correction is to be included.
DIMFG	Dimension flag; this flag consists of two numbers which specify the dimensions of the augmented observation matrix, H_A, state vector $\underline{x}_A$, covariance matrices P_A, P'_A, M_A and the gain matrix K_A. Its value is determined in initialization based upon the input values of BSFG, TRFG, HSFG and SSFG.
EUFG	Ephemeris unit conversion flag; designates whether ephemeris information has to be unit converted. $$\text{EUFG} = \begin{cases} 0, & \text{no unit conversion} \\ 1, & \text{unit conversion} \end{cases}$$

3.1.3.2 Definition of Flags (con't)

Symbol		Description
HSFG		Horizon sensor flag; indicates whether a horizon sensor is to be included as a source of measurement data $\text{HSFG} = \begin{cases} 0, & \text{no horizon sensor} \\ 1, & \text{horizon sensor} \end{cases}$
INF		Internal state transition initialization flag for all conics $\text{INF} \begin{cases} = 0 & \text{no total state transition initialization is needed} \\ \neq 0 & \text{total state transition initialization is needed (see paragraph 3.4.2.1)} \end{cases}$ Set INF $\neq$ 0 1. at the beginning of the case 2. after an orbit rectification is made
IUFG		Input unit conversion flag; indicates whether input has to be unit converted $\text{IUFG} = \begin{cases} 0, & \text{no unit conversion} \\ 1, & \text{unit conversion} \end{cases}$
K(P)R		Orbit-rectification flag; serves dual purpose of flag that defines simulation runs for which orbit rectification capability is to be included and a constant used to determine when accuracy in estimate of position is adequate for rectification purposes
m	M	Conic flag; determines conic
MASK		Computer word-length simulation flag; when non-zero, this flag causes a pseudo-shortened word length to be used in computations. Value assigned to flag indicates the number of bits of $P_A(t_k)$, $K_i(t_k)$, $\hat{\underline{x}}(t_k)$ that are to be set equal to zero
MINFG		Star selection flag; defines the method by which a star is selected for the space sextant. NEEDED ONLY WHEN SSFG = 1 $\text{MINFG} = \begin{cases} 0, & \text{prespecified input star} \\ 1, & \text{optimum ideal star without practical limitations} \\ 2, & \text{optimum realistic star based on a given star catalog} \end{cases}$

3.1.3.2 Definition of Flags (con't)

Symbol	Description
OUFG	Output unit conversion flag; indicates whether output has to be unit converted OUFG = { 0, no unit conversion; 1, unit conversion
PEMSFG	Electromagnetic sensor output print flag
PGIDFG	Guidance output print flag; defines guidance quantities that are desired in the output print
PLINFG	Linear approximation output print flag
PNAVFG	Navigation output print flag
PRFG	Determines desirability of computing actual trajectory PRFG = { 0, actual ≡ nominal; 1, computes actual
PRINFG	Output print flag; this flag establishes the desirability and frequency of printing the output
PROFG	Internal conic transfer flag PROFG = 1 no conic transfer took place between observations PROFG = 4 a conic transfer did take place between standard observations (see paragraph 3.4.2.2.3 Block II. 3.2) Set by time advance section of calculations control.
PSTRFG	State transition output print flag; defines state transition matrices that are desired in the output print
PTRJFG	Trajectory output print flag; defines trajectory block quantities desired in the output print
RAFG	Internal actual initialization flag for current conic RAFG = 0 no actual initialization needed for current conic RAFG ≠ 0 actual initialization is needed for current conic (see paragraph 3.4.3.1) Set ≠ 0 1. at the beginning of each conic 2. after a velocity correction is made

3.1.3.2 Definition of Flags (con't)

Symbol	Description
RECTFG	Internal flag designating that an orbit rectification was performed
RNFG	Internal nominal and state transition initialization flag for for current conic

$$\text{RNFG} \begin{cases} = 0 & \text{no nominal and state transition initialization needed for current conic} \\ \neq 0 & \text{nominal and state transition initialization is needed for current conic (see paragraph 3.4.1.1 and 3.4.2.1)} \end{cases}$$

Set $\neq 0$

1. at the beginning of each conic
2. after an orbit rectification has been made

RVFG	Conic state flag; determines source of initial conditions for start of each conic

$$\text{RVFG} = \begin{cases} 0, & \text{position and velocity at start of each conic is input} \\ 1, & \text{position and velocity at start of each conic except first is computed internally} \end{cases}$$

SQRTFG	Square-root Kalman filter flag; defines the computational procedure to be used for the state vector estimation. When non-zero, the covariance matrices of the measurement noise should be diagonal

$$\text{SQRTFG} = \begin{cases} 0, & \text{standard Kalman filter formulation} \\ 1, & \text{square-root version of Kalman filter} \end{cases}$$

SSFG	Space sextant flag; indicates whether a space sextant is to be included as a source of measurement data.

$$\text{SSFG} = \begin{cases} 0, & \text{no space sextant} \\ 1, & \text{space sextant} \end{cases}$$

3.1.3.2 Definition of Flags (con't)

Symbol	Description
TERFG	Terminal constraint flag; defines the terminal constraints for which statistics (i.e., T_E) must be computed.

$$\text{TERFG} = \begin{cases} 0, & \text{no terminal constraints} \\ 1, & \underline{r}(t_A) \\ 2, & r(t_A),\ \gamma(t_A),\ \theta(t_A) \\ 3, & r(t_A),\ \gamma(t_A),\ \underline{u}(t_A) \end{cases}$$

Symbol	Description
TERMFG	Termination flag; internally set flag used to terminate a simulation run
TRFG	Ground tracker flag; when non-zero, ground tracking information is to be included as a source of measurement data

$$\text{TRFG} = \begin{cases} 1, & \text{one tracker} \\ 2, & \text{two trackers} \\ 3, & \text{three trackers} \end{cases}$$

Symbol	Description
VELFG	Velocity correction flag; determines the guidance law that is used to compute midcourse corrections

$$\text{VELFG} = \begin{cases} 0, & \text{no velocity correction} \\ 1, & \underline{r},\ \Delta T = 0 \quad \text{TERFG} = 1 \\ 2, & \underline{r},\ \Delta T \neq 0 \quad \text{TERFG} = 1 \\ 3, & r,\ \gamma,\ \theta,\ \Delta T = 0 \quad \text{TERFG} = 2 \\ 4, & r,\ \gamma,\ \theta,\ \Delta T \neq 0 \quad \text{TERFG} = 2 \\ 5, & r,\ \gamma,\ \underline{u},\ \Delta T \neq 0 \quad \text{TERFG} = 3 \end{cases}$$

Symbol	Description
ζ_i	Internally set instrument flag; ζ_i refers to the i^{th} instrument

$$i = \begin{cases} 1, & \text{ground tracker No. 1} \\ 2, & \text{ground tracker No. 2} \\ 3, & \text{ground tracker No. 3} \\ 4, & \text{horizon sensor} \\ 5, & \text{space sextant} \end{cases}$$

At each t_k when ζ_i is zero, the i^{th} instrument shall be omitted from consideration in the NAVIGATION BLOCK

3.1.3.2 Definition of Flags (con't)

Symbols	Description
ζ_o	Internally set navigation flag; when ζ_o is zero at t_k, the NAVIGATION BLOCK is not exercised (i.e., no measurements are made).

3.1.3.3 Definition of Symbols

The quantities used in the flow charts and equations are defined below. The input form and dimension of these quantities are also given where appropriate. The notation given in the first column corresponds to the notation used in the equations, whereas the notation in the second column corresponds to the mnemonic symbols as used in the input print format.

Symbol		Description
a		Semi-transverse (semi-major) axis of a conic
a_i	A(IL)	Range rate variance constants where
b_i	B(I)	$\sigma^2_{\dot{\rho}_i} \overset{Df}{=} a_{io} + a_{i1}(1 + b_i\dot{\rho}_i)^2 \rho_i + a_{i2}(1 + b_i\dot{\rho}_i)^2 \rho_i^2 + a_{i3}(1 + b_i\dot{\rho}_i)^4$ ρ_i and $\dot{\rho}_i$ are the range and range rate of the spacecraft relative to the i^{th} tracker
$\underline{b}$		The p-dimensional vector of bias errors in measurements data. (The vector $\underline{b}$ will be considered in terms of sub-vectors $\underline{b}_i$ where the $\underline{b}_i$ is related to the i^{th} instrument. This distinction will carry over to other quantities such as K_A, H_A.)

3.1.3.3 Definition of Symbols (con't)

Symbol		Description
b_{ik}	B(IK)	Range variance constants where $\sigma^2_{\rho i} \stackrel{Df}{=} b_{io} + b_{i1}\rho_i^2 + b_{i2}\rho_i^4$
B		The (p x p) dimensional covariance matrix associated with $\underline{b}$ (a symmetric matrix)
B_i	B(I)	Covariance of bias errors in tracker number i. This (7 x 7) matrix can be partitioned into the form (see below)
B_{ioI}	B(I)O	Diagonality of B_{iI} for i = 1, 2, 3. Value of zero indicates that B_{iI} is diagonal matrix.
B_{ioL}	B(I)OL	Diagonality of B_{iL}. Value of zero indicates that B_{iL} is diagonal matrix

$$B_i = \begin{bmatrix} B_{IL} & 0 \\ 0 & B_{iI} \end{bmatrix}$$

where B_{iI} is (4 x 4) and represents the covariance matrix of the bias errors in the tracker measurements. B_{iL} is (3 x 3) and represents the covariance matrix of the tracker location uncertainty. Elements of B_{iI} are ordered in terms of ρ_i, $\dot{\rho}_i$, ψ_i, η_i. The elements of B_{iL} are ordered in terms of x_{iT}, y_{iT}, z_{iT} (i.e., the components of the position vector to the tracker at t_o). The B_i matrix will be input as follows:

i. Six numbers will be input to define B_{iL} and ten numbers will be input to define B_{iI}. Symmetry of the matrices will establish the remaining elements.

ii. The first 3 elements of B_{iL} designate the diagonal elements. The remaining 3 are input on the same row when $B_{ioI} = 1$. The first 4 elements of B_{iI} are the diagonal elements. The remaining 6 elements are input when $B_{ioL} = 1$.

3.1.3.3 Definition of Symbols (con't)

Symbol		Description
B_4	B(4)	Covariance of bias errors in horizon sensor. It is a (3 x 3) symmetric matrix. Six elements of this matrix are input in a manner analogous to that of B_{iL}. The remaining elements are established from symmetry considerations. The elements of B_4 are ordered in terms of α, δ, and β^H.
B_{4O}	B(4)O	Designates diagonality of B_4. Value of zero indicates that B_4 is a diagonal matrix.
B_5	B(5)	Covariance of bias error in the space sextant.
C_{i_j}	CIJ	Covariance matrix constant; multiplies computed covariance matrix; i = I = 1, 2, 3; j = J = 1, 2, . . . , 10.
C_T	CT	Time conversion factor; converts units of input time to units used in ephemeris
$\overline{d^2(t_k)}$		Trace of $D(t_k)$
$D(t_k)$		Covariance matrix of the error in the estimated velocity correction
e		Orbit eccentricity
E		Eccentric anomaly
EPL		Ephemeris length conversion factor, equal to ephemeris unit of length/internal unit of length
EPT		Ephemeris time conversion factor, equal to ephemeris unit of time/internal unit of time
F		Hyperbolic eccentric anomaly (i. e., hyperbolic equivalent of E)
F(t)		The (6 x 6) dimensional instantaneous state transition matrix where $\Phi(t_k, t_j)$ is the solution of

$$\frac{d}{dt} \Phi(t, t_j) = F(t) \Phi(t, t_j) \; ; \; \Phi(t_j, t_j) = I$$

3.1.3.3 Definition of Symbols (con't)

Symbol		Description
$\underline{g}(t_A)$	$\begin{bmatrix} G1*(TA) \\ G2*(TA) \\ G3*(TA) \end{bmatrix}$	Nominal acceleration vector at target
$\underline{h}(t_k)$		Angular momentum vector, $\underline{h} \overset{Df}{=} \underline{R} \times \underline{V}$
$H(t_k)$		The m x 6 observation matrix. For linear perturbation theory, $H(t_k)$ arises through the assumption that $\underline{y}(t_k) \approx H(t_k)\, \underline{x}(t_k)$
$H_A(t_k)$		The m x (p + 6) dimensional augmented observation matrix. In terms of partitioned matrices $H_A(t_k) = [H(t_k)\ H_B]$ Thus, $\underline{z}(t_k) \approx H_A(t_k)\, \underline{x}_A(t_k) + \underline{v}(t_k)$
H_B		The (m x p) bias matrix. The bias errors are assumed to effect the measurements according to $H_B\underline{b}$
$H_i(t_k)$		The m^i x(p + 6) dimensional observation matrix for the i^{th} aiding instrument, $\sum_{i=1}^{k} m^i = m.$
IUL		Input length conversion factor, equal to input unit of length/ internal unit of length
IUT		Input time conversion factor, equal to input unit of time/ internal unit of time.
$K(t_k)$		The (6 x m) optimal gain matrix
$K_A(t_k)$		The (p x 6) x m augmented optimal gain matrix
K_E	KE	Kepler's equation accuracy constant for elliptic orbits. Built in as 1.0×10^{-7}.
K_F	KF	Kepler's equation accuracy constant for hyperbolic orbits. Built in as 1.0×10^{-7}.

3.1.3.3 Definition of Symbols (con't)

Symbol		Description
$K_i(t_k)$		The $(p + 6) \times m^i$ i^{th} instrument optimal gain matrix
K_p^R	K(P)R	Orbit rectification position constant. Serves as criterion for defining when position estimate is adequate for rectification purposes
K_v^R	K(V)R	Orbit rectification velocity constant; serves as criterion for defining when velocity estimate is adequate for rectification purposes.
m		Internal symbol used to designate dimensionality of certain quantities
m	MF	Number of conics. There may be ten possible conics, $m = MF = 1, 2, \ldots, 10$
$M(t_k)$	M(TK)	Covariance of perturbation state vector (6 x 6) dimensional covariance matrix associated with $\underline{x}(t_k)$
$M(t_o)$	M	Covariance of initial state $\overset{Df}{=} E[\underline{x}(t_o)\, \underline{x}^T(t_o)]$. It is a (6 x 6) symmetrix matrix of which only 21 elements are input. The remaining 15 should be established using the symmetry. $M(t_o)$ will be input according to the following schedule:

i. The first six inputs designate the diagonal elements, M_{ii} or MII $i = I = 1, 2, \ldots, 6$

ii. The remaining fifteen elements are input by now when $M_{oo} = 1$ in the order
$M_{12}, M_{13}, \ldots, M_{16}$
$M_{23}, M_{24}, \ldots, M_{26}$
$M_{34}, M_{35}, M_{36}, M_{45}, M_{46}, M_{56}$

$M_A(t_k)$ — Covariance of augmented perturbation state vector $(p + 6) \times (p + 6)$ covariance matrix associated with $\underline{x}_A(t_k)$

$$M_A(t_k) \overset{Df}{=} \begin{bmatrix} M(t_k) & M_1(t_k) \\ M_1^T(t_k) & M_2(t_k) \end{bmatrix}$$

where $M_1(t_k)$ is (6 x p) matrix and $M_2(t_k)$ is (p x p) matrix.

3.1.3.3 Definition of Symbols (con't)

Symbol		Description
M_{oo}	MOO	Defines nature of $M(t_o)$
		$$M_{oo} = \begin{cases} 0, & M(t_o) \text{ is diagonal} \\ 1, & M(t_o) \text{ is non-diagonal} \end{cases}$$
n		Internal mean angular motion $n^2 \overset{Df}{=} \mu_\ell / a^3$
n_1		Internal mean angular motion $n_1^2 \overset{Df}{=} \mu_\ell / -a^3$
$n^2(t_k)$		Trace of $N(t_k)$
$N(t_k)$		Covariance of the actual velocity correction
NO. STARS		Number of stars to be used in a star catalog
OUL		Output length conversion factor, equal to output unit of length/internal unit of length
OUT		Output time conversion factor, equal to output unit of time/ internal unit of time
p		Internal symbol used to designate dimensionality of certain quantities
$P(t_k)$		Covariance of error in estimate $\hat{\underline{x}}(t_k)$ of $\underline{x}(t_k)$, (6 x 6)
		$$P(t_k) \overset{Df}{=} E\{[\hat{\underline{x}}(t_k) - \underline{x}(t_k)][\hat{\underline{x}}(t_k) - \underline{x}(t_k)]^T\}$$
$P'(t_k)$		Covariance of error in estimate $\hat{\underline{x}}'(t_k)$ of $\underline{x}(t_k)$, (p + 6) x (p + 6)
$P_A(t_k)$		Covariance of error in estimate $\hat{\underline{x}}_A(t_k)$ of $\underline{x}_A(t_k)$, (p + 6) x (p + 6)
		$$P_A(t_k) \overset{Df}{=} \begin{bmatrix} P(t_k) & P_1(t_k) \\ P_1^T(t_k) & P_2(t_k) \end{bmatrix}$$
		where $P_1(t_k)$ is (6 x p) and $P_2(t_k)$ is (p x p)

3.1.3.3 Definition of Symbols (con't)

Symbol		Description
$Q(t_k)$		(6 x 6) diagonal matrix to account for the computational and dynamic model errors, i.e., $$P(t_k) = \Phi(t_k, t_{k-1})\, P'(t_{k-1})\, \Phi^T(t_k, t_{k-1}) + Q(t_k)$$
$\underline{r}^*(t_A^*)$	$\begin{bmatrix} X1^*(TA) \\ X2^*(TA) \\ X3^*(TA) \end{bmatrix}$	Nominal position vector at target
$r(t_o^m)$	ROT(M)	Maximum permissible distance from central body to space-craft. NEEDED ONLY WHEN RVFG = 1.
r_{iT}	SMR(IT)	Distance from i^{th} tracker to center of body; i = 1, 2, 3
r_ℓ	SR(L)	Radius of body ℓ
$R(t_k)$		The (m x m) dimensional covariance matrix associated with the m-dimensional measurement noise vector
$\underline{R}(t_k)$		(3 x 1) position vector at t_k
$\underline{R}(T_k)$		(3 x 1) position vector at $T_k = T_L + C_T t_k$
$R_i(t_k)$	RI(TK)	Covariance matrix of noise in ground trackers; i = I = 1, 2, 3 $$R_i(t_k) \overset{Df}{=} C_{ij} \begin{bmatrix} \sigma^2_{\rho_i} & \sigma_{\rho_i \rho_i} & \sigma_{\rho_i \psi_i} & \sigma_{\rho_i \eta_i} \\ & \sigma^2_{\rho_i} & \sigma_{\rho_i \psi_i} & \sigma_{\rho_i \eta_i} \\ \text{Symmetric} & & \sigma^2_{\psi_i} & \sigma_{\psi_i \eta_i} \\ & & & \sigma^2_{\eta_i} \end{bmatrix}$$
R_{io}	RI(0)	Defines nature of $R_i(t_k)$; i = 1, 2, 3 $$R_{io} = \begin{cases} 0, & R_i(t_k) \text{ is diagonal} \\ 1, & R_i(t_k) \text{ is non-diagonal} \end{cases}$$

3.1.3.3 Definition of Symbols (con't)

Symbol		Description
R_v	R(V)	Variance ratio $$R_v \overset{Df}{=} \frac{\overline{d^2(t_k)}}{\overline{\Delta V^2(t_k)}}$$
$R_{v_{min}}$	R(VMIN)	Minimum variance ratio. When the variance ratio R_v becomes less than $R_{v_{min}}$, a velocity correction may occur. Dimensionless constant.
$R_4(t_k)$	R4(TK)	Covariance of noise in horizon sensor measurements. This is a (3 x 3) matrix of which 6 elements are input as a tabulated function of time. First three elements are diagonal elements. Ordered in terms of α, δ, β^H. The elements are input as a function of time with the format:

Time or TIME	$r^4_{11}(t_k)$	$r^4_{22}(t_k)$	$r^4_{33}(t_k)$
TIME	R4(11)	R4(22)	R4(33)

and

Time or TIME	$r^4_{12}(t_k)$	$r^4_{13}(t_k)$	$r^4_{23}(t_k)$
TIME	R4(12)	R4(13)	R4(23)

Provision is made for 50 different values of the time argument

$R_{40}(t_k)$	R4(O)	Defines nature of $R_4(t_k)$. Table of 50 values $$R_{40}(t_k) = \begin{cases} 0, & R_4(t_k) \text{ is diagonal} \\ 1, & R_4(t_k) \text{ is non-diagonal} \end{cases}$$
$R_5(t_k)$	R5(TK)	Covariance of noise in space sextant measurements. Tabulated as a function of time (50 values)
Λ^*		(3 x 3) rotation matrix. Matrix accomplishes transformation from an auxiliary in-plane coordinate system to the basic non-rotating coordinate frame.

3.1.3.3 Definition of Symbols (con't)

Symbol		Description
$\underline{s}(t_k)$		(3 x 1) star vector
STAR		Denotes number of star to be used for the $t_{k-1} \leq t \leq t_k$ in star catalog defined by CATFG. NEEDED ONLY WHEN MINFG = 0
t_A	TA	Time of arrival. This is time at which program is terminated and represents the time at which the target is reached
t_k (or t_k^m)		The k^{th} observation time ($k = 0, 1, \ldots, N$). The superscript m is sometimes included to indicate the conic section ($m = 1, 2, \ldots, 10$). The subscript k is reset equal to zero at the start of each conic
t_o^m	T(M)	Time at start of m^{th} conic
t_f^m		Final time in m^{th} conic
T		Transformation matrix between the terminal state vector and the terminal constraints
T_E		Terminal constraint covariance matrix $$T_E(t_k) \stackrel{Df}{=} T(t_A^*)\ \Phi(t_A^*, t_k)\ P(t_k)\ \Phi^T(t_A^*, t_k)\ T^T(t_A^*)$$
T_L	TL	Julian date corresponding to $t = t_o$. This is a double precision input number (i.e., 16 digits)
$\underline{u}$		Unit vector perpendicular to trajectory plane
$\underline{v}^*(t_A)$	$\begin{bmatrix} X4*(TA) \\ X5*(TA) \\ X6*(TA) \end{bmatrix}$	Nominal velocity vector at target
$\underline{v}(t_k)$		The m-dimensional vector of uncorrelated (between sampling times) measurement noise
$V(t_k)$		Covariance matrix of the estimated velocity correction, (3 x 3)
W		(3 x 3) or (4 x 4) diagonal weighting matrix used in the optimum star selection

3.1.3.3 Definition of Symbols (con't)

Symbol		Description
$\underline{x}(t_k)$		The six-dimensional perturbation state vector $$\underline{x}(t_k) \overset{Df}{=} \underline{X}(t_k) - \underline{X}^*(t_k)$$
$\hat{\underline{x}}(t_k)$		Best linear estimate of $\underline{x}(t_k)$ based on measurement data $\underline{z}(t_k)$. This is a (6 x 1) vector
$\hat{\underline{x}}'(t_k)$		Best linear estimate of $\underline{x}(t_k)$ based on measurement data $\underline{z}(t_{k-1})$
$\underline{x}_A(t_k)$		The (p + 6) dimensional augmented perturbation state vector. In terms of partitioned vectors $$\underline{x}_A(t_k) \overset{Df}{=} \begin{bmatrix} \underline{x}(t_k) \\ \underline{b}(t_k) \end{bmatrix}$$
$\hat{\underline{x}}_A(t_k)$		Best linear estimate of $\underline{x}_A(t_k)$ based on measurement data $\underline{z}(t_k)$ $$\hat{\underline{x}}_A(t_k) \overset{Df}{=} \begin{bmatrix} \hat{\underline{x}}(t_k) \\ \hat{\underline{b}}(t_k) \end{bmatrix}$$
$\underline{X}(t_k) \overset{Df}{=} \begin{bmatrix} \underline{R}(t_k) \\ \underline{V}(t_k) \end{bmatrix}$		The six-dimensional state vector. The first three components denote position and the last three represent velocity. The nominal state is written $\underline{X}^*(t_k)$.
$\tilde{\underline{x}}(t_k)$		The (6 x 1) error in the estimate vector $$\tilde{\underline{x}}(t_k) \overset{Df}{=} \underline{x}(t_k) - \hat{\underline{x}}(t_k)$$
$\underline{X}(t_o)$	$\underline{X}$(T1)	Actual trajectory initial state vector (6 x 1). NEEDED ONLY WHEN PRFG = 1, ACTFG = 0.
$\underline{X}^*(t_o^m)$	$\underline{X}$*(TM)	Nominal initial state vector for m^{th} conic. This is a (6 x 1) vector whose first three components [X1*(TM), X2*(TM), X3*(TM)] represent the position vector and the last three components [X4*(TM), X5*(TM), X6*(TM) represent the velocity.

3.1.3.3 Definition of Symbols (con't)

Symbol		Description
$\underline{y}(t_k)$		The m-dimensional perturbed observation vector $$\underline{y}(t_k) \overset{Df}{=} \underline{Y}(t_k) - \underline{Y}^*(t_k)$$
$\underline{Y}(t_k)$		The m-dimensional observation vector. Components of this vector represent quantities that can be measured with a physical device (e.g., ground radar or space sextant). This vector contains no random errors.
$\underline{Y}_i(t_k)$		The m^i-dimensional i^{th} instrument observation vector
$\underline{z}(t_k)$		The m-dimensional vector of actual measurement data $$\underline{z}(t_k) \overset{Df}{=} \underline{y}(t_k) + H_B\,\underline{b} + \underline{v}(t_k)$$
$\underline{z}_i(t_k)$		The m^i-dimensional vector of i^{th} instrument actual measurement data
α		Internal local vertical horizon sensor angle
β^H		Horizon sensor half-subtended angle
β^H_{max}	BET(MAX)	Maximum allowable subtended angle, (degrees)
β^H_{min}	BET(MIN)	Minimum allowable subtended angle, (degrees)
β_{ℓ}		Internal half-subtended angle by body ℓ
β^S		Internal space sextant half-subtended angle
$\overline{\gamma^2}$	GAM(SQ)	Variance of velocity correction direction error, (radians)^2
δT		Change in t_A due to velocity correction
Δt_k	OBS.DELT	Interval between observations. Δt_k is used for a pre-specified number of observations according to the following schedule:

Number of Observations	Δt_k
⋮	⋮

3.1.3.3 Definition of Symbols (con't)

Symbol		Description
$\underline{\Delta V}(t_k)$		(3 x 1) actual velocity correction vector. This represents correction actually applied to trajectory
$\hat{\underline{\Delta V}}(t_k)$		(3 x 1) velocity correction vector based on $\hat{\underline{x}}(t_k)$
$\overline{\Delta V^2(t_k)}$		Trace of $V(t_k)$
$\Delta \underline{x}(t_k)$	SM DEL X	The (6 x 1) linear approximation to the state vector $\Delta \underline{x}(t_k) \overset{Df}{=} \underline{x}(t_k) - \Phi(t_k, t_o)\, \underline{x}(t_o)$
$\Delta \underline{y}(t_k)$		The linear approximation to the observation vector $\Delta \underline{y}_i(t_k) \overset{Df}{=} \underline{y}_i(t_k) - H_{iT1}\, \underline{x}(t_k),\ i = 1, 2, 3$; $\Delta y_4(t_k) \overset{Df}{=} \underline{y}_4(t_k) - [H_H(t_k);\ 0]\, \underline{x}(t_k)$; $\Delta y_5(t_k) \overset{Df}{=} y_5 - [H_5\ ;\ 0]\, \underline{x}(t_k)$
$\epsilon(t_o^m)$	ROL(M)	Tolerance on $r(t_o^m)$
ϵ_ℓ	EPS(L)	Tolerance on the half-subtended angle of the planet ℓ, $\ell = 0, 1, \ldots, 6$
ϵ_{po}	EPS(PO)	Criterion to establish whether the vehicle is behind the central body, i.e., cosine of angle between vehicle and Earth vectors with respect to the central body, usually ≤ 0.
ϵ_{so}	EPS(SO)	Criterion to establish whether the Sun is in the line of sight of the vehicle from the Earth.
ϵ_w	EPS(W)	Input constant restricting the allowable stars to a specific part of the sky
$\overline{\eta^2}$	ETA(SQ)	Variance of velocity correction magnitude error. Dimensionless constant.
η_i	ETA(I)	Internal tracker azimuth angle; $i = 1, 2, 3$

3.1.3.3 Definition of Symbols (con't)

Symbol		Description
Θ_i	TH(I)	Longitude of i^{th} tracker, units of degrees; $i = 1, 2, 3$
$\varkappa$	KAP	Velocity correction cutoff time. When $t_k > \varkappa t_A$, no velocity correction is permitted
$\varkappa_i$	KAP(I)	Terminal constraint accuracy constants. When $\sqrt{T_{E_{ii}}} < \varkappa_i$, no velocity correction is permitted; $i = I = 1, 2, 3, 4$
$\varkappa_s$	KAP(S)	Space sextant half-subtended angle multiplicative constant; i.e., $\varkappa_s \beta_s$. When $\varkappa_s = 0$, the simulated angle measured by the space sextant is from the center of the planet to the star.
$\Lambda(t_k)$		(3 x 6) guidance matrix $$\underline{\Delta \hat{V}}(t_k) \overset{Df}{=} \Lambda(t_k)\, \hat{\underline{x}}'(t_k)$$
μ_ℓ	MU(L)	Gravitational constant of body $$\ell = L = \begin{cases} 0, & \text{Earth} \\ 1, & \text{Moon} \\ 2, & \text{Sun} \\ 3, & \text{Venus} \\ 4, & \text{Mars} \\ 5, & \text{Saturn} \\ 6, & \text{Jupiter} \end{cases}$$
ν		True anomaly
$\Pi(t_k)$	PI	The factored $P(t_k)$ matrix $$P(t_k) = \Pi(t_k)\, \Pi^T(t_k)$$ (The definitions of $\Pi'(t_k)$, $\Pi_A(t_k)$, $\Pi'_A(t_k)$ should be obvious in light of the definitions above.)
ρ	RHO	Tracker range
$\underline{\rho}$	RHO VECTOR	Tracker range vector (3 x 1)

3.1.3.3 Definition of Symbols (con't)

Symbol		Description
$\dot{\rho}$	RHO DOT	Tracker range rate
$\dot{\underline{\rho}}$	RHO D VECTOR	Tracker range rate vector (3 x 1)
ρ^1_{max}	RHO(1MAX)	Maximum possible range for range measurement
ρ^2_{max}	RHO(2MAX)	Maximum possible range for angle measurements
σ^i	SIG(I)	A priori statistics error. Constant that allows the effect of incorrect a priori statistics to be examined $(1 + \sigma^i)$, $R_{jj}(t_k)$ is used to generate noise vector $\underline{v}_i(t_k)$ whereas $R_i(t_k)$ is used to determine estimates, $i = I = 1, 2, \ldots, 5$.
σ^γ	SIG(GAM)	A priori statistics error in the velocity correction direction
σ^η	SIG(ETA)	A priori statistics error in the velocity correction magnitude
$\sigma^2_{\eta_i}$	SETISQ	Variance of noise in measurement of azimuth angle η_i, units of $(rad)^2$
$\sigma_{\rho_i \eta_i}$	SRIETI	Covariance of noise in range ρ_i and azimuth angle η_i measurement; units of (length) x (rad)
$\sigma_{\dot{\rho}_i \eta_i}$	SRDIEI	Covariance of noise in range rate $\dot{\rho}_i$ and azimuth angle η_i measurements, units of (length/time) x (rad)
$\sigma_{\rho_i \dot{\rho}_i}$	SRIRDI	Covariance of noise in range ρ_i and range rate $\dot{\rho}_i$ measurements
$\sigma_{\rho_i \psi_i}$	SRIPSI	Covariance of noise in range ρ_i and elevation angle ψ_i measurements, units of (length)(rad)
$\sigma_{\dot{\rho}_i \psi_i}$	SRDIPI	Covariance of noise in range rate $\dot{\rho}_i$ and elevation angle ψ_i measurements; units of (length/time) x (rad)
$\sigma^2_{\psi_i}$	SPSISQ	Variance of noise in measurement of elevation angle ψ_i, units of $(rad)^2$; $i = I = 1, 2, 3$
$\sigma_{\psi_i \eta_i}$	SPIEI	Covariance of noise in elevation angle ψ_i and azimuth angle η_i measurements, units of $(rad)^2$

3.1.3.3 Definition of Symbols (con't)

Symbol		Description
$\tau(t_k^m)$		Internal time from pericentron passage in the m^{th} conic corresponding to t_k
$\tau(t_o^m)$		Internal time from pericentron passage at the start of the m^{th} conic
φ_i	PH(I)	Latitude of i^{th} tracker in degrees; $i = 1, 2, 3$
$\Phi(t_k, t_j)$		The (6 x 6) dimensional state transition matrix relating the states $\underline{x}(t_k)$ and $\underline{x}(t_j)$
$\Phi_A(t_k, t_j)$		The (p + 6) x (p + 6) dimensional augmented state transition matrix. In terms of partitioned matrices $\Phi_A(t_k, t_j) = \begin{bmatrix} \Phi(t_k, t_j) & O_1 \\ O_1^T & I \end{bmatrix}$ where O_1 is (6 x p) matrix of zeros
ψ_i		Internal tracker elevation angle; $i = 1, 2, 3$
ψ_{io}	PSI(0)	Minimum permissible elevation angle, units of degrees (the maximum angle is built in as 1.536 radians)
ω	OME	Earth's rotation rate (degrees/time)

3.2 FUNCTIONAL ORGANIZATION OF THE PROGRAM

The diagram that immediately follows these paragraphs is designated as the Level I flow chart. It does nothing other than summarize the basic structure of the program in terms of the basic functional operations that must be performed. It can be considered as consisting of two types of functions. First, operations that constitute the basic computational cycle; these functions are described by the blocks that have been designated with Roman numerals. The details relative to these blocks can be found in Paragraph 3.4. Blocks A, B, and C describe functions that either occur once, i.e., Block B), are required in order to make the program operate meaningfully (i.e., Block A), or act passively relative to the computational cycle (i.e., Block C). These three blocks are described in Paragraph 3.2

The INPUT block represents a summary of the quantities that an engineer must input. No computations are contained within this block. In the GENERAL INITIALIZATION block, computations that must be performed once during a specific simulation run and/or logical decisions that must be made for proper operation within the basic computational cycle are accomplished. The OUTPUT block defines the quantities that are to be available for print-out purposes and contains computations that are not required in the basic computational cycle. This program uses guidance and navigation policies that are based upon the techniques of linear perturbation theory. To apply these methods, it is necessary to compute a nominal (or reference) trajectory. This task is accomplished in the TWO-BODY INTERPLANETARY NOMINAL block. Based upon the nominal trajectory, it is possible to compute appropriate state transition matrices which provide one of the corner stones for all linear guidance and navigation policies. These matrices are computed within the EXPLICIT STATE TRANSITION block. The actual state of the spacecraft is simulated in the TWO-BODY ACTUAL block.

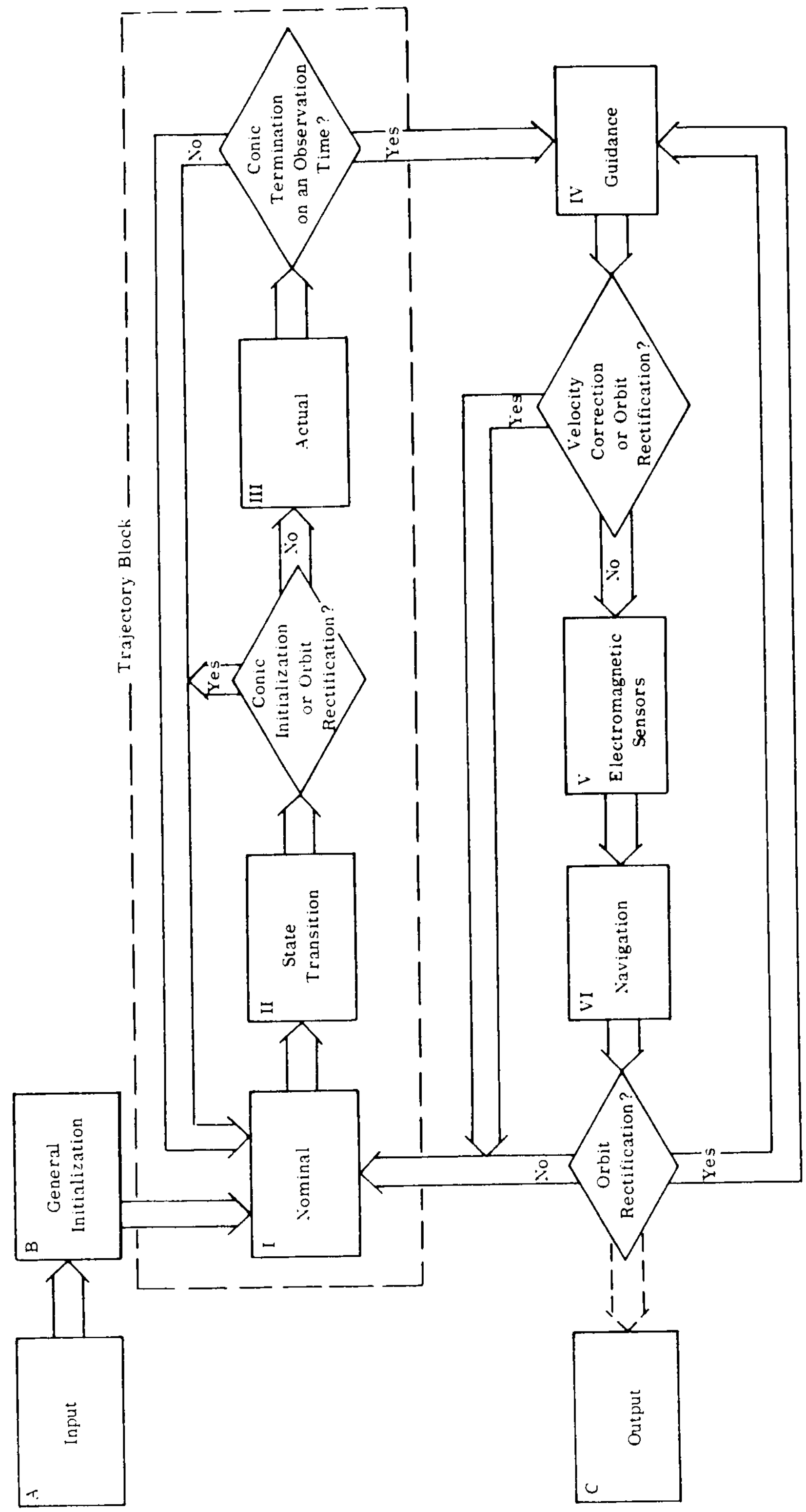
During a simulation run, it is necessary to compute the state transition matrix between the current time t_k and the terminal time t_A^* for use in the GUIDANCE block. In order to accomplish this, it is necessary at the start of the simulation (i.e., at t_o) to compute $\Phi(t_A^*, t_o)$. This procedure is referred to as CONIC INITIALIZATION and involves only Blocks I and II. The transition matrix $\Phi(t_A^*, t_k)$ must be recomputed whenever the nominal trajectory is modified. The nominal is changed in order that the hypothesis regarding the linear relation between the nominal and actual trajectories remains valid. This occurrence is referred to as ORBIT RECTIFICATION. The decision regarding the time at which a rectification is to be introduced is determined in the NAVIGATION block.

The basic schedule of observation times is input by the engineer. It can happen that the time at which a conic is terminated may not coincide with an observation time. When this occurs, the trajectory calculations are performed without entering the "feed-back" branch that involves Blocks IV, V, and VI.

The restriction has been incorporated into the program that the times at which velocity corrections occur and at which observational data is obtained are mutually exclusive. For this reason the GUIDANCE block appears before the NAVIGATION and ELECTROMAGNETIC SENSOR blocks. When a velocity correction is deemed necessary, these two blocks are bypassed. The computations relating to the advisability of a correction and the correction itself are contained in the GUIDANCE block.

The ELECTROMAGNETIC SENSOR block provides simulations of several instruments (e.g., ground-based radar, horizon sensor, space sextant) that are used to obtain information relating to the state of the spacecraft. The measurement data from these instruments are computed for both the nominal and actual trajectories.

The NAVIGATION block uses the measurement data generated in Block V in order to obtain a new estimate of the state. The statistics regarding the covariance of the error in the estimate are used to determine when an orbit rectification should be introduced. When rectification is called for, the flow must return to the GUIDANCE block so that initial conditions can be generated which will permit the new nominal to satisfy the same terminal constraints.



Level I Flow Chart - Guidance and Navigation for Interplanetary Space Missions

3.3 INPUT, GENERAL INITIALIZATION AND OUTPUT

For a specific simulation run, the logical and computational flow within the program is established to a large extent by the values assigned to several input "flags". These flags are defined in Paragraph 1.3.2. In paragraph 1.3.3 all the symbols and input quantities are defined and the form of their input explained where appropriate. The definitions contain in the first column the symbols as used in the equations and in the second column the corresponding mnemonic symbols as used in the input print format.

The most general input print format is stated in Paragraph 3.1.1. In Paragraph 3.1.2 the capability of converting the input data to a set of internal program units is described. The input quantities and the blocks which require them are listed in Paragraph 3.1.3. There are no flow charts associated with the input block.

The detailed flow charts and equations of the logical and computational actions performed only once during a specific simulation run are described in the GENERAL INITIALIZATION block, Paragraph 3.3.2. Level II and Level III flow charts are included in this block were appropriate.

Definitions of the quantities which are required as the outputs of each of the functional blocks are given in Paragraphs 1.3.2 and 1.3.3. Among those quantities, several of them are required either for print or in other functional blocks. Those quantities that may be required for print-out are given by the most general output print format in Paragraph 3.3.1.5. The program does contain several output formats. The specific format desired is established by the values assigned to the output print flags. These flags and the associated quantities to be printed are described in Paragraph 3.3.1.1 Level II and Level III are included to describe the computations that are required in the OUTPUT block. In Paragraph 3.3.1.2 the coordinate conversion option is described. Paragraph 3.3.1.3 describesthe output computations and Paragraph 3.3.1.4 defines the output unit conversion option.

3.3.1 Input - Block A

3.3.1.1 Input Print Format

CASE C PERFCRMANCE ASSESSMENT CF AICEC-INERTIAL GUICANCE AND NAVIGATICN SYSTEMS FCR FREE FALL

**** INPUT CATA ****

IDENTIFICATICN CARD 1

IDENTIFICATICN CARD 2

TRAJECTCRY INPUT ****

TL = C.   TA = C.   TO = C.   CT = 0.

MF = 1C   RVFG = 1

RL(I) =	C.	C.	0.	0.	C.	C.	C.
MU(I) =	0.	C.	C.	0.	0.	0.	C.

	CONIC TIMES	CENTRFG	RCL	RCT
T(1) =	C.	C	C.	C.
T(2) =	C.	C	0.	C.
T(3) =	0.	0	0.	0.
T(4) =	0.	C	C.	0.
T(5) =	C.	C	C.	C.
T(6) =	C.	0	0.	C.
T(7) =	0.	0	0.	C.
T(8) =	0.	0	0.	0.
T(9) =	C.	0	0.	C.
T(1C) =	C.	C	0.	0.

INITIAL NOMINAL CONIC STATES

X*(T 1) VECTOR =	C.	0.	C.	0.	0.	0.
X*(T 2) VECTOR =	C.	C.	C.	0.	0.	0.
X*(T 3) VECTOR =	C.	C.	0.	C.	0.	C.
X*(T 4) VECTOR =	C.	0.	0.	C.	0.	C.
X*(T 5) VECTOR =	C.	0.	C.	C.	0.	0.
X*(T 6) VECTOR =	C.	C.	C.	0.	0.	0.
X*(T 7) VECTOR =	C.	C.	0.	0.	0.	0.
X*(T 8) VECTOR =	0.	0.	0.	0.	0.	0.
X*(T 9) VECTOR =	C.	C.	C.	C.	0.	0.
X*(T1C) VECTOR =	C.	C.	0.	0.	0.	0.

PRFG = C   ACTFG = 0

X(T1) VECTOR =	C.	C.	0.	0.	0.	C.

INPUT FCR GUICANCE BLOCK ****

CASE C PERFCRMANCE ASSESSMENT CF AIDED-INERTIAL GUICANCE AND NAVIGATION SYSTEMS FCR FREE FALL

**** INPUT DATA ****

TERFG = 0 VELFG = C VAR ETA SQ = 0. VAR GAM SQ = C. R(VMIN) = C.

SIG(ETA) = 0. SIG(GAM) = 0.

KAPPA = 0. KAPPA (I) = 0. 0. 0. 0.

X*(TA) VECTCR = C. 0. 0. 0. 0. 0.

G*(TA) VECTOR = C. 0. 0.

INPUT FCR ELECTROMAGNETIC SENSCRS BLOCK ****

BSFG = 1 SSFC = 1 HSFG = 1 TRFG = 3 MINFG = 0 NCSTAR = 0

KAP(S) = 0. EPS(W) = C.

EPS(L) = C. 0. 0. 0. C. C. C.

GROUND TRACKING DATA-

EPS(PC) = C. EPS(SC) = C. CME = 0. RHC 1 MAX = 0. RHO 2 MAX = C.

SMALL R MAGS = 0. 0. C. PHI (I) = 0. 0. 0.

THETA (I) = 0. 0. C. PSIO (I) = C. C. C.

CCVARIANCE MATRIX CCNSTANTS (1ST CCLUMN = TIME)

10 BY 4 MATRIX C(I,J)

	1	2	3	4
1	C.	0.	C.	0.
2	C.	C.	0.	0.
3	C.	0.	0.	0.
4	C.	0.	0.	0.
5	C.	C.	C.	0.
6	C.	0.	0.	0.
7	0.	C.	0.	0.
8	0.	0.	0.	0.
9	C.	C.	C.	0.
1C	C.	0.	0.	0.

R1(0) = 0 R2(C) = 0 R3(C) = 0

RANGE RATE VARIANCE CCNSTANTS

3 BY 4 MATRIX SMALL A(I,L)

1	2	3	4

CASE 0 PERFCRMANCE ASSESSMENT OF AIDEC-INERTIAL GUICANCE AND NAVIGATION SYSTEMS FCR FREE FALL

**** INPUT CATA ****

	1	2	3	4
1	0.	0.	0.	0.
2	C.	0.	C.	0.
3	C.	C.	0.	0.

SMALL B(I) =, 0. 0. 0.

RANGE VARIANCE CONSTANTS

3 BY 3 MATRIX SMALL B(I,K)

	1	2	3
1	C.	C.	C.
2	C.	C.	C.
3	C.	C.	C.

SIGMA(PSI SCR)	=	0.	0.	C.
SIGMA(ETA SCR)	=	0.	0.	0.
SIGMA(PSI ETA)	=	C.	0.	0.
SIGMA(RHCDCT PSI)	=	C.	C.	0.
SIGMA(RHO RHCDCT)	=	C.	C.	C.
SIGMA(RHC PSI)	=	0.	C.	C.
SIGMA(RHC ETA)	=	C.	0.	0.
SIGMA(RHODOT ETA)	=	0.	0.	0.

HCRIZCN SENSCR-

R4(0) = 1 BETA(MIN) = 0. BETA(MAX) = 0.

	TIME	R4(11)	R4(22)	R4(33)
T 1	C.	C.	C.	C.
T 2	C.	C.	C.	C.
T 3	C.	C.	C.	C.
T 4	0.	C.	C.	C.
T 5	C.	C.	C.	C.
T 6	C.	C.	C.	C.
T 7	C.	C.	C.	C.
T 8	C.	C.	C.	C.
T 9	0.	C.	0.	0.
T1C	C.	C.	C.	C.

	TIME	R4(12)	R4(13)	R4(23)
T 1	C.	C.	C.	C.
T 2	C.	C.	C.	C.
T 3	C.	C.	0.	0.
T 4	C.	C.	0.	0.
T 5	C.	C.	C.	C.
T 6	C.	C.	C.	C.
T 7	C.	C.	C.	0.

CASE 0 PERFCRMANCE ASSESSMENT OF AIDED-INERTIAL GUIDANCE ANC NAVIGATICN SYSTEMS FOR FREE FALL

**** INPUT CATA ****

T 8	C.	C.	C.	O.
T 9	C.	C.	C.	C.
T10	C.	C.	C.	O.

SPACE SEXTANT ANC/CR HCRIZCN SENSOR

	TIME	BODY	R(5)	STAR
T 1	C.	0	C.	0
T 2	C.	0	O.	C
T 3	C.	0	C.	C
T 4	C.	0	C.	0
T 5	C.	0	C.	0
T 6	C.	0	C.	C
T 7	O.	0	O.	0
T 8	C.	0	C.	0
T 9	C.	0	C.	C
T10	C.	0	C.	C

INPUT FCR NAVIGATICN BLCCK ****

PCFG = C SCRTFG = C MASK =CCCCCCCCCCC K(P)R = O. K(V)R = C. MCC = 0

SIG (I) = C. C. O. C. C.

6 BY 6 CIAGCNAL MATRIX M(TK)
O. C. O. O. O. C.

CIAGCNAL C MATRIX INPUT TABLE

	TIME	Q(11)	Q(22)	Q(33)	Q(44)	Q(55)	Q(66)
T 1	O.	C.	O.	O.	O.	C.	C.
T 2	C.	O.	O.	O.	C.	C.	O.
T 3	O.	C.	O.	C.	O.	O.	O.
T 4	O.	C.	C.	C.	O.	O.	C.
T 5	C.	C.	C.	O.	C.	C.	C.
T 6	C.	O.	O.	C.	C.	C.	O.
T 7	O.	O.	O.	C.	O.	O.	O.
T 8	C.	C.	C.	C.	O.	O.	C.
T 9	C.	C.	O.	O.	C.	C.	C.
T1C	C.	O.	O.	C.	C.	O.	O.

B(1)O = 0 B(2)O = C B(3)O = C B(4)C = 0 B(1)CL = 0 B(2)OL = C B(3)CL = C

4 BY 4 CIAGONAL MATRIX B(II)
O. O. O. O.

CASE C PERFCRMANCE ASSESSMENT OF AIDED-INERTIAL GUIDANCE ANC NAVIGATICN SYSTEMS FCR FREE FALL

**** INPUT CATA ****

4 BY 4 CIAGCNAL MATRIX B(2I)

0.	0.	0.	C.

4 BY 4 CIAGONAL MATRIX B(3I)

C.	C.	C.	0.

3 BY 3 CIAGCNAL MATRIX B(4)

0.	C.	C.

B(5) = C.

3 BY 3 CIAGCNAL MATRIX B(1L)

0.1CCCCCCCE C1	C.	0.

3 BY 3 CIAGONAL MATRIX B(2L)

0.	C.	0.

3 BY 3 CIAGCNAL MATRIX B(3L)

0.	0.	C.

BIAS NCISE-

B1 BAR VECTCR =	C.	0.	C.	C.	C.	C.	C.
B2 BAR VECTCR =	C.	C.	C.	0.	0.	C.	0.
B3 BAR VECTOR =	0.	0.	0.	0.	0.	0.	0.
B4 BAR VECTCR =	C.	0.	0.				
B5 BAR =	C.						

PRINT FLAGS ****

PTRJFG = 0 PSTRFG = C PGIDFG = 0 PEMSFG = C PNAVFG = 0 PLINFG = 0 CCORDFG = C

CUFG = C CUL = C. CUT = C.

IUFG = C IUL = C. IUT = 0.

EUFG = C EPL = 0. EPT = 0.

NUMBER CF OBSERVATIONS	OBSERVATICN DELTAS	PRINFG
0	0.	C
C	0.	C
0	0.	0
C	C.	C
0	0.	0
0	0.	0
0	0.	0
C	C.	C
0	0.	C
0	0.	0

3.3.1.2 Input Unit Conversion

A capability to convert the input data into internal units is available in the program. The following set of conversion factors is employed for this purpose.

IUL = input unit of length/internal unit of length

IUT = input unit of time/internal unit of time

The quantities which are converted with their conversion factors are given below. The outline follows the layout given in the input format 3.3.1.1. The internal data will be denoted by a superscript id.

TL; KE; KF; RVFG; N/A (not applicable)

$$\left[R(0), \ldots\ldots, R(7)\right]^{id} = (IUL)^{-1} \left[R(0) \ldots\ldots R(7)\right]$$

$$\left[MU(0), \ldots\ldots, MU(7)\right]^{id} = (IUT)^{2} (IUL)^{-3} \left[MU(0), \ldots\ldots, MU(7)\right]$$

$$\left[TA, TO\right]^{id} = (IUT)^{-1} \left[TA, TO\right]$$

$$\left[T(1), \ldots\ldots, T(10)\right]^{id} = (IUT)^{-1} \left[T(1), \ldots\ldots, T(10)\right]$$

C(1), ..., C(10) N/A

$$\left[ROL(1), \ldots\ldots, ROL(10)\right]^{id} = (IUL)^{-1} \left[ROL(1), \ldots\ldots, ROL(10)\right]$$

$$\left[ROT(1), \ldots\ldots, ROT(10)\right]^{id} = (IUL)^{-1} \left[ROT(1), \ldots\ldots, ROT(10)\right]$$

$$\begin{bmatrix} X1^*(T1), \ldots\ldots, X1^*T(10) \\ X2^*(T1), \ldots\ldots, X2^*T(10) \\ X3^*(T1), \ldots\ldots, X3^*T(10) \end{bmatrix}^{id} = (IUL)^{-1} \begin{bmatrix} X1^*(T1), \ldots\ldots, X1^*T(10) \\ X2^*(T1), \ldots\ldots, X2^*T(10) \\ X3^*(T1), \ldots\ldots, X3^*T(10) \end{bmatrix}$$

$$\begin{bmatrix} X4^*(T1), \ldots\ldots, X4^*(T10) \\ X5^*(T1), \ldots\ldots, X5^*(T10) \\ X6^*(T1), \ldots\ldots, X6^*(T10) \end{bmatrix}^{id} = (IUT)(IUL)^{-1} \begin{bmatrix} X4^*(T1), \ldots\ldots, X4^*(T10) \\ X5^*(T1), \ldots\ldots, X5^*(T10) \\ X6^*(T1), \ldots\ldots, X6^*(T10) \end{bmatrix}$$

PRFG; ACTFG; N/A

$$[\mathrm{ROT}(1), \ldots\ldots, \mathrm{ROT}(10)]^{id} = (\mathrm{IUL})^{-1}[\mathrm{ROT}(1), \ldots\ldots, \mathrm{ROT}(10)]$$

$$\begin{bmatrix} X1^*(T1), & \ldots\ldots, & X1^*T(10) \\ X2^*(T1), & \ldots\ldots, & X2^*T(10) \\ X3^*(T1), & \ldots\ldots, & X3^*T(10) \end{bmatrix}^{id} = (\mathrm{IUL})^{-1} \begin{bmatrix} X1^*(T1), & \ldots\ldots, & X1^*T(10) \\ X2^*(T1), & \ldots\ldots, & X2^*T(10) \\ X3^*(T1), & \ldots\ldots, & X3^*T(10) \end{bmatrix}$$

$$\begin{bmatrix} X4^*(T1), & \ldots\ldots, & X4^*(T10) \\ X5^*(T1), & \ldots\ldots, & X5^*(T10) \\ X6^*(T1), & \ldots\ldots, & X6^*(T10) \end{bmatrix}^{id} = (\mathrm{IUT})(\mathrm{IUL})^{-1} \begin{bmatrix} X4^*(T1), & \ldots\ldots, & X4^*(T10) \\ X5^*(T1), & \ldots\ldots, & X5^*(T10) \\ X6^*(T1), & \ldots\ldots, & X6^*(T10) \end{bmatrix}$$

PRFG; ACTFG; N/A

$$\begin{bmatrix} X1(T1) \\ X2(T1) \\ X3(T1) \end{bmatrix}^{id} = (\mathrm{IUL})^{-1} \begin{bmatrix} X1(T1) \\ X2(T1) \\ X3(T1) \end{bmatrix} \quad ; \quad \begin{bmatrix} X4(T1) \\ X5(T1) \\ X6(T1) \end{bmatrix}^{id} = (\mathrm{IUT})(\mathrm{IUL})^{-1} \begin{bmatrix} X4(T1) \\ X5(T1) \\ X6(T1) \end{bmatrix}$$

Note, when ACTFG = 1 $\underline{X}(T1)$ is not converted.

TERFG; VELFG; ETA(SQ); GAM(SQ); RV(MIN); KAP; N/A

$$\begin{bmatrix} KAP(1) \\ KAP(2) \\ KAP(3) \end{bmatrix}^{id} = (\mathrm{IUL})^{-1} \begin{bmatrix} KAP(1) \\ KAP(2) \\ KAP(3) \end{bmatrix} \quad \text{when TERFG} = 1$$

KAP(2); KAP(3); KAP(4) N/A when TERFG = 2, 3

$$\begin{bmatrix} X1^*(TA) \\ X2^*(TA) \\ X3^*(TA) \end{bmatrix}^{id} = (\mathrm{IUL})^{-1} \begin{bmatrix} X1^*(TA) \\ X2^*(TA) \\ X3^*(TA) \end{bmatrix} \quad ; \quad \begin{bmatrix} X4^*(TA) \\ X5^*(TA) \\ X6^*(TA) \end{bmatrix}^{id} = (\mathrm{IUT})(\mathrm{IUL})^{-1} \begin{bmatrix} X4^*(TA) \\ X5^*(TA) \\ X6^*(TA) \end{bmatrix}$$

$$\begin{bmatrix} G1^*(TA) \\ G2^*(TA) \\ G3^*(TA) \end{bmatrix}^{id} = (\mathrm{IUT})^2 (\mathrm{IUL})^{-1} \begin{bmatrix} G1^*(TA) \\ G2^*(TA) \\ G3^*(TA) \end{bmatrix}$$

BSFG; SSFG; HSFG; TRFG; MINFG; CATFG; NOSTAR; EPS(PO); EPS(SO) N/A

$$(\mathrm{OME})^{id} = (\mathrm{IUT})\ \mathrm{OME}$$

$$\begin{bmatrix} RHO(1MAX) \\ RHO(2MAX) \end{bmatrix}^{id} = (\mathrm{IUL})^{-1} \begin{bmatrix} RHO(1MAX) \\ RHO(1MAX) \end{bmatrix}$$

$[R(1T), R(2T), R(3T)]^{id} = (IUL)^{-1} [R(1T), R(2T), R(3T)]$

PH(1), PH(2), PH(3), TH(1), TH(2); TH(3); PS1(0); PS2(0); PS3(0) N/A

$[T1, \ldots\ldots, TN]^{id} = (IUT)^{-1} [T1, \ldots\ldots, TN]$

CIJ $I = 1, 2, 3$ $J = 1, \ldots, 10$ N/A

R1(0); R2(0); R3(0) N/A

$[B(1), B(2), B(3)]^{id} = (IUL)(IUT)^{-1} [B(1), B(2), B(3)]$

$[B(10), B(20), B(30)]^{id} = (IUL)^{-2} [B(10), B(20), B(30)]$

[B(11), B(21), B(31)] N/A

$[B(12), B(22), B(32)]^{id} = (IUL)^{2} [B(12), B(22), B(32)]$

$[A(10), A(20), A(30)]^{id} = (IUT)^{2}(IUL)^{-2} [A(10), A(20), A(30)]$

$[A(11), A(21), A(31)]^{id} = (IUT)^{2}(IUL)^{-1} [A(11), A(21), A(31)]$

$[A(12), A(22), A(32)]^{id} = (IUT)^{2} [A(12), A(22), A(32)]$

$[A(13), A(23), A(33)]^{id} = (IUT)^{2}(IUL)^{-2} [A(13), A(23), A(33)]$

$$\left.\begin{array}{l} \text{SPS1SQ, SPS2SQ, SPS3SQ,} \\ \text{SET1SQ, SET2SQ, SET3SQ,} \\ \text{SP1E1 , SP2E2 , SP3E3,} \end{array}\right\} \quad \text{N/A}$$

$[SRD1P1, SRD2P2, SRD3P3]^{id} = (IUT)(IUL)^{-1} [SRD1P1, SRD2P2, SRD3P3]$

$[SRD1E1, SRD2E2, SRD3E3]^{id} = (IUT)(IUL)^{-1} [SRD1E1, SRD2E2, SRD3E3]$

$[SR1RD1, SR2RD2, SR3RD3]^{id} = (IUT)(IUL)^{-2} [SR1RD1, SR2RD2, SR3RD3]$

$$\begin{bmatrix} SR1PS1, & SR2PS2, & SR3PS3 \\ SR1ET1, & SR2ET2, & SR3ET3 \end{bmatrix}^{id} = (IUL)^{-1} \begin{bmatrix} SR1PS1, & SR2PS2, & SR3PS3 \\ SR1ET1, & SR2ET2, & SR3ET3 \end{bmatrix}$$

R4(0); BET(MIN); BET(MAX) N/A

$$\left.\begin{array}{l}\text{R4(11); R4(22); R4(33); R4(12); R4(13); R4(23)}\\ \text{BODY; STAR; R(5); POFG; SQRTFG; MASK}\\ \text{K(P)R; K(V)R; MOD; SIG1, 2, 3, 4, 5}\end{array}\right\} \text{N/A}$$

$$\begin{bmatrix}\text{M11, M12, M33}\\ \text{M12, M13, M23}\end{bmatrix}^{id} = (\text{IUL})^{-2}\begin{bmatrix}\text{M11, M12, M33}\\ \text{M12, M13, M23}\end{bmatrix}$$

$$\begin{bmatrix}\text{M44, M55, M66}\\ \text{M45, M46, M56}\end{bmatrix}^{id} = (\text{IUT})^{2}(\text{IUL})^{-2}\begin{bmatrix}\text{M44, M55, M66}\\ \text{M45, M46, M56}\end{bmatrix}$$

$$\begin{bmatrix}\text{M14, M15, M16}\\ \text{M24, M25, M26}\\ \text{M34, M35, M36}\end{bmatrix}^{id} = (\text{IUT})(\text{IUL})^{-1}\begin{bmatrix}\text{M14, M15, M16}\\ \text{M24, M25, M26}\\ \text{M34, M35, M36}\end{bmatrix}$$

B(1)O; B(1)OL; B(2)O; B(2)OL; B(3)O; B(3)OL: B(4)O N/A

$$[\text{B(IL) MATRICES}]^{id} = (\text{IUL})^{-2}\ \text{B(IL)} \quad \text{I} = 1,2,3$$

$$[\text{B(1I)11, B(2I)11, B(3I)11}]^{id} = (\text{IUL})^{-2}\ [\text{B(1I)11, B(2I)11, B(3I)11}]$$

$$[\text{B(1I)22, B(2I)22, B(3I)22}]^{id} = (\text{IUT})^{2}(\text{IUL})^{-2}\ [\text{B(1I)22, B(2I)22, B(3I)22}]$$

$$\left.\begin{array}{l}\text{B(1I)33, B(1I)44, B(1I)34}\\ \text{B(2I)33, B(2I)44, B(2I)34}\\ \text{B(3I)33, B(3I)44, B(3I)34}\end{array}\right\} \text{N/A}$$

$$\begin{bmatrix}\text{B(1I)13, B(1I)14, B(2I)13, B(2I)14}\\ \text{B(3I)13, B(3I)14}\end{bmatrix}^{id} = (\text{IUL})^{-1}\begin{bmatrix}\text{B(1I)13, B(1I)14, B(2I)13}\\ \text{B(2I)14, B(3I)13, B(3I)14}\end{bmatrix}$$

$$\begin{bmatrix}\text{B(1I)23, B(1I)24, B(2I)23}\\ \text{B(2I)24, B(3I)23, B(3I)24}\end{bmatrix}^{id} = (\text{IUT})(\text{IUL})^{-1}\begin{bmatrix}\text{B(1I)23, B(1I)24, B(2I)23}\\ \text{B(2I)24, B(3I)23, B(3I)24}\end{bmatrix}$$

BIAS NOISE N/A

$$[\text{OBS. DELT}]^{id} = (\text{IUT})^{-1}\ [\text{OBS. DELT}]$$

The information obtained from the ephemeris routine ANTR1 is also converted to internal units. The conversion factors used are

EPL ephemeris length/internal unit of length

EPT ephemeris time/internal unit of time

Thus,

$$[\text{PLNT L POS VEC}]^{id} = (\text{EPL})^{-1} [\text{PLNT L POS VEC}]$$

$$[\text{PLNT L VEL VEC}]^{id} = (\text{EPT})(\text{EPL})^{-1} [\text{PLNT L VEL VEC}]$$

$$L = 0, 1, \ldots, 6$$

3.3.1.3 Blocks Where Input Quantities are Required

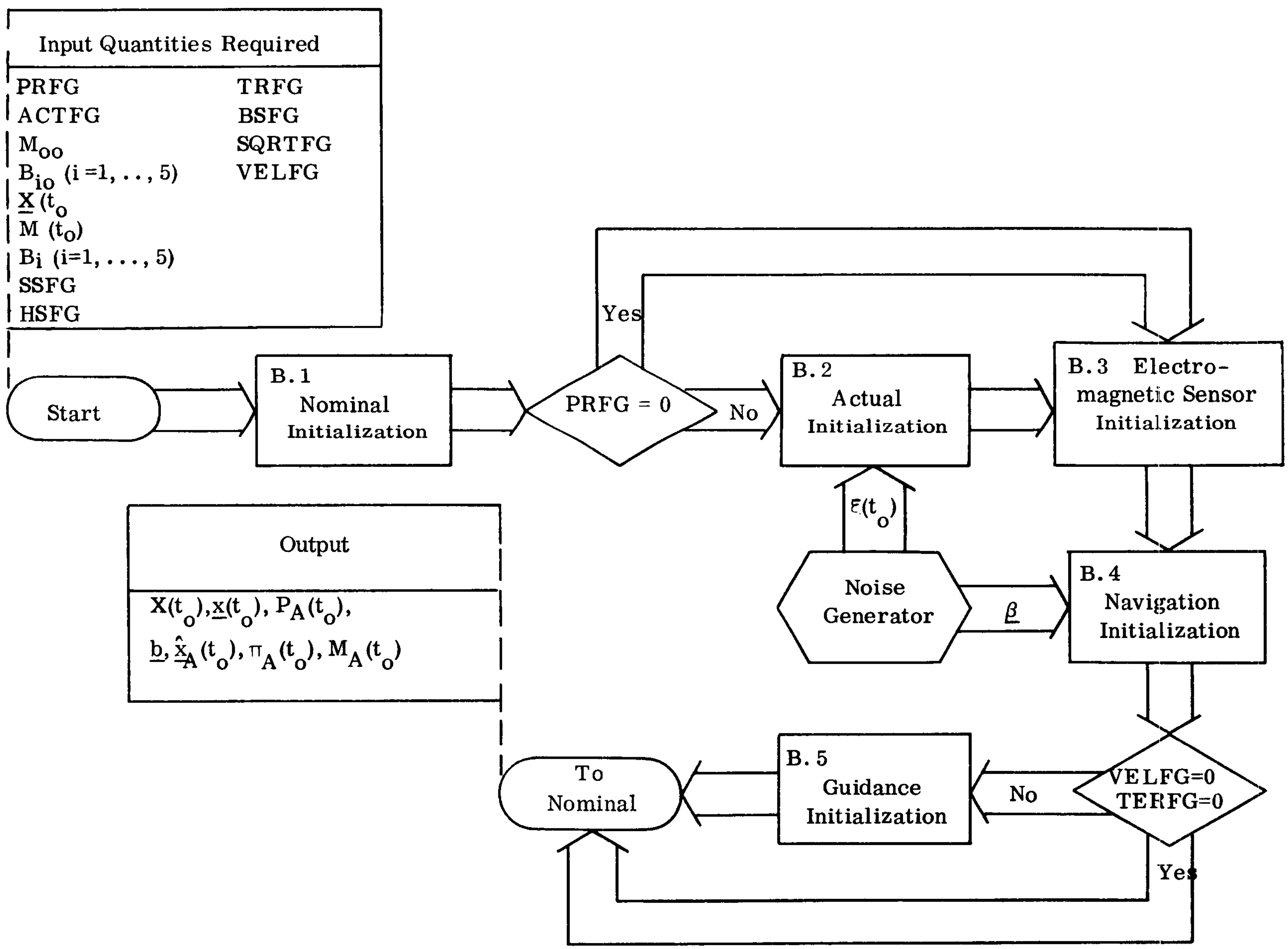
INPUT CATEGORY	QUANTITY	WHERE USED
NOMINAL	TRAJFG	Nominal
	RVFG	Nominal
	t_A	Nominal, State Transition
	T_L	Actual, Electromagnetic Sensors
	C_T	Actual, Electromagnetic Sensors
	K_E	Nominal, Actual
	K_F	Nominal, Actual
	m	Nominal, State Transition
	r_ℓ	Electromagnetic Sensors
	μ_ℓ	Nominal, Actual
	IUL	Input
	IUT	Input
	EPL	Actual, Electromagnetic Sensors
	EPT	Actual, Electromagnetic Sensors
	t_o^m	Nominal, State Transition, Actual
	CENTRFG(m)	Nominal, Electromagnetic Sensors
	$r(t_o^m)$	Nominal
	$\epsilon(t_o^m)$	Nominal
	$\underline{X}^*(t_o^m)$	General Initialization, Nominal, Actual

INPUT CATEGORY	QUANTITY	WHERE USED
ACTUAL	ACTFG	General Initialization
	PRFG	General Initialization, Electromagnetic Sensors
	$\underline{X}(t_o)$	General Initialization, Actual
GUIDANCE	VELFG	Guidance
	TERFG	General Initialization
	$\varkappa$, $\varkappa_i$	Guidance
	R_{vmin}	Guidance
	$\bar{\eta}^2$, $\bar{\gamma}^2$	Guidance
	$\underline{r}(t_A)$, $\underline{v}(t_A)$, $\underline{g}(t_A)$	General Initialization, Guidance
ELECTROMAGNETIC SENSORS	BSFG	General Initialization, Electromagnetic Sensors
	SSFG	General Initialization, Electromagnetic Sensors
	HSFG	General Initialization, Electromagnetic Sensors
	TRFG	General Initialization, Electromagnetic Sensors
	MINFG	Electromagnetic Sensors
	CATFG	Electromagnetic Sensors
	r_{iT}	Electromagnetic Sensors
	φ_i	Electromagnetic Sensors
	θ_i	Electromagnetic Sensors
	ψ_{io}	Electromagnetic Sensors
	$\sigma^2_{\psi i}$, $\sigma^2_{\eta i}$, ... $\sigma_{\psi i \eta i}$	Electromagnetic Sensors

INPUT CATEGORY	QUANTITY	WHERE USED
ELECTROMAGNETIC SENSORS	c_{ij}, a_i , b_i, b_{ik}	Electromagnetic Sensors
	ω	Electromagnetic Sensors
	ϵ_{po}, ϵ_{so}	Electromagnetic Sensors
	ρ^1_{max}, ρ^2_{max}	Electromagnetic Sensors
	R_{io}	Electromagnetic Sensors
	R_i	Electromagnetic Sensors, Navigation
	R_{40}	Electromagnetic Sensors
	R_4	Electromagnetic Sensors, Navigation
	β^H_{min}, β^H_{max}	Electromagnetic Sensors
	R_5	Electromagnetic Sensors, Navigation
	NO STAR	Electromagnetic Sensors
	BODY	Electromagnetic Sensors
	STAR	Electromagnetic Sensors
NAVIGATION	POFG	General Initialization, Navigation
	SQRTFG	General Initialization, Navigation, Guidance
	MASK	Navigation
	K^R_p	Navigation
	K^R_v	Navigation
	σ_i	General Initialization
	ϵ	Navigation
	M_{oo}	General Initialization

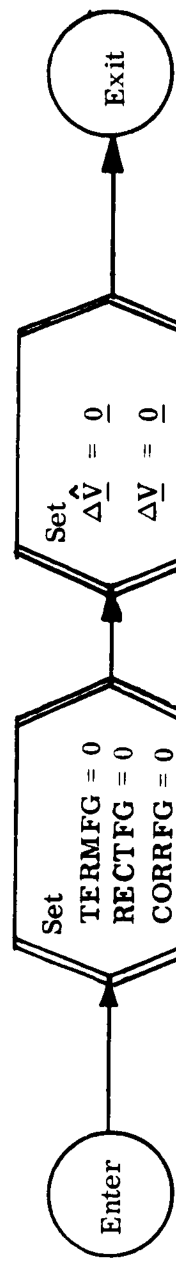
INPUT CATEGORY	QUANTITY	WHERE USED
NAVIGATION	$M(t_o)$	General Initialization
	B_{ioI}	General Initialization
	B_{iL}	General Initialization
	B_{ioL}	General Initialization
	B_{iL}	General Initialization
	B_{40}	General Initialization
	B_4	General Initialization
	B_5	General Initialization
OUTPUT	PRINFG	Output
	PTRJFG	Output
	PSTRFG	Output
	PGIDFG	Output
	PEMSFG	Output
	PNAVFG	Output
	PLINFG	Output
	COORDFG	Output
	Δt_k	Nominal, State Transition, Actual
	OUL	Output
	OUT	Output

3.3.2 General Initialization - Block B

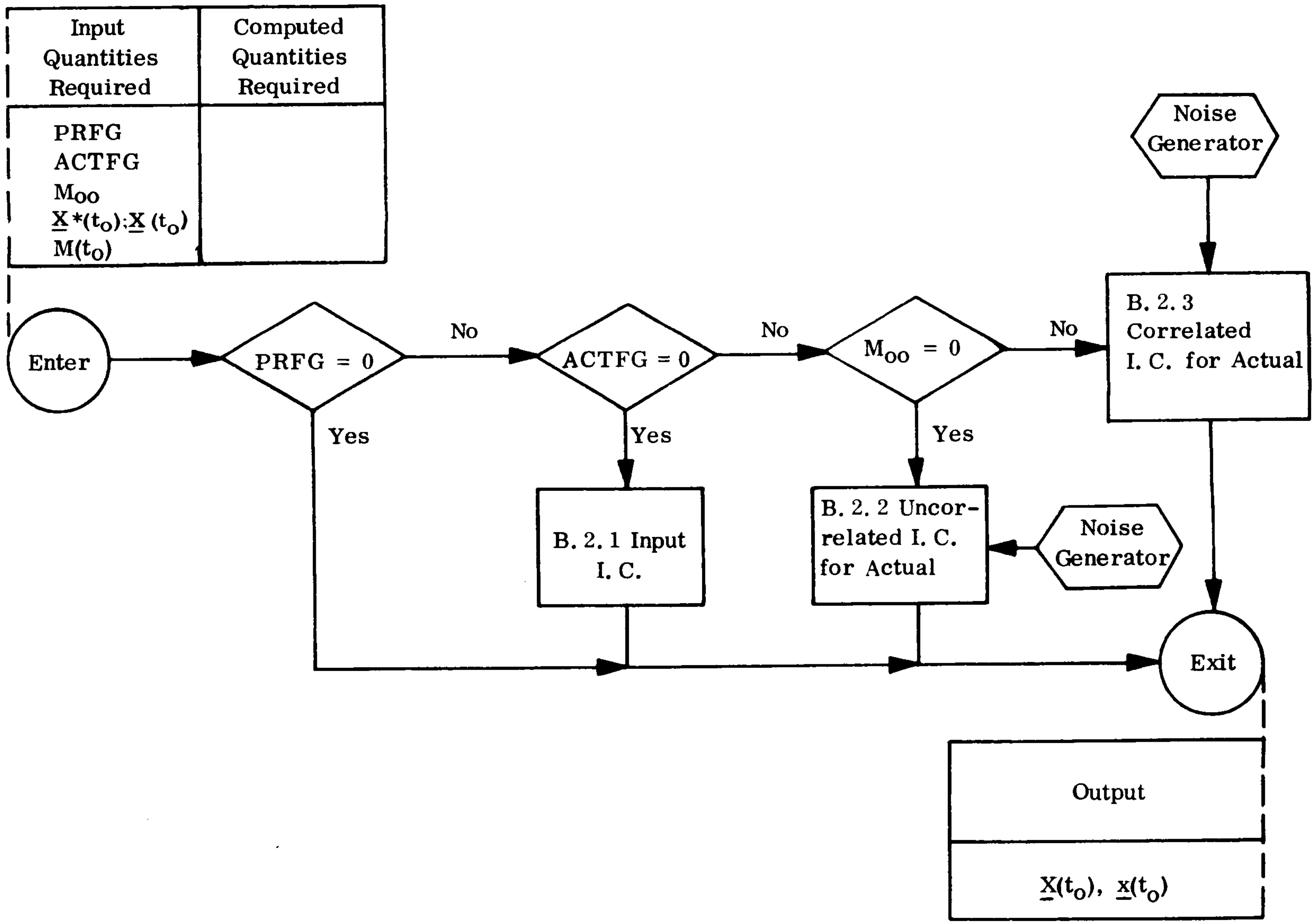


3.3.2.1 Level II Flow Chart - General Initialization

3.3.2.2 Detailed Flow Charts and Equations



3.3.2.2.1 Nominal Initialization - Block B.1



3.3.2.2.2 Actual Initialization - Block B.2

B. 2. 1 Input I. C.

$$\underline{X}(t_o) \text{ is input}$$

B. 2. 2 Uncorrelated I. C. for Actual

Generate six numbers $x_i(t_o)$ according to a Gaussian distribution, each with mean zero and with variance $M_{ii}(t_o)$ $(i = 1, \dots, 6)$.

Let these numbers form the vector

$$\underline{x}(t_o) = \begin{bmatrix} x_1(t_o) \\ \vdots \\ x_6(t_o) \end{bmatrix}$$

Form the initial conditions for the actual trajectory

$$\underline{X}(t_o) = \underline{X}^*(t_o) + \underline{x}(t_o)$$

B. 2. 3 Correlated I. C. for Actual

Generate a triangular matrix such that

$$\underline{x}(t_o) = T_M(t_o)\, \underline{\xi}(t_o)$$

where

$$M(t_o) = T_M(t_o)\, D_M(t_o)\, T_M^T(t_o)$$

$$D_M(t_o) = \text{diagonal matrix with elements } D_{M_{ii}}(t_o)$$

Generate six gaussian random numbers $\xi_i(t_o)$ with mean zero and variance $D_{M_{ii}}(t_1)$, $(i = 1, 2, \dots, 6)$. The means of accomplishing this transformation is defined elsewhere.

Form

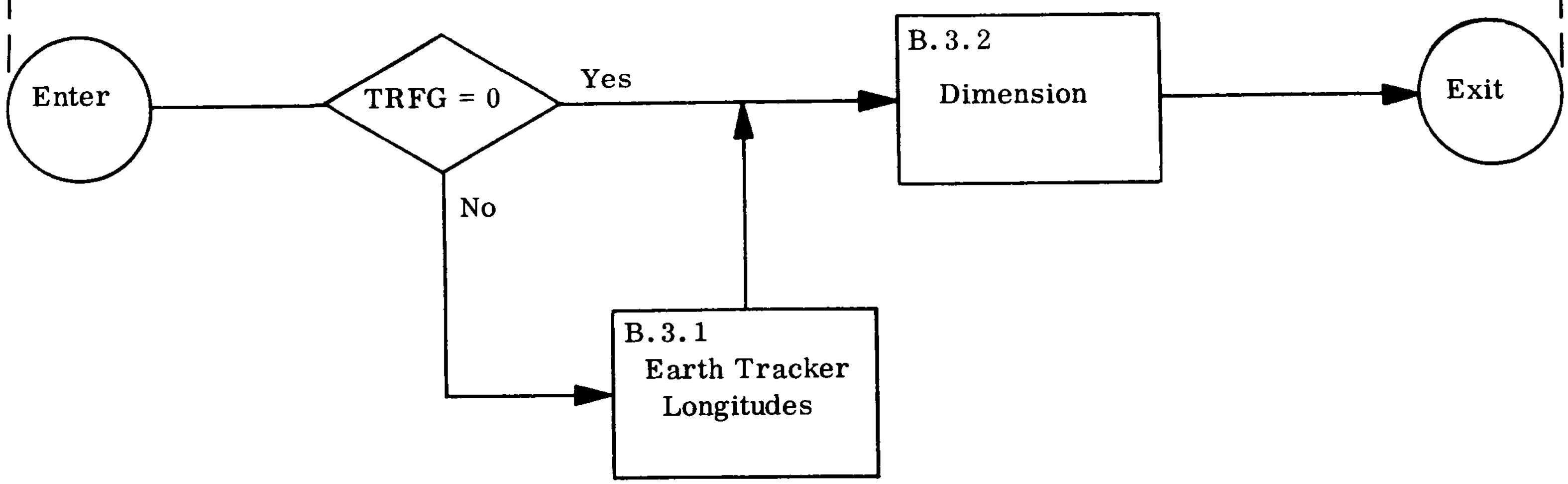
$$\underline{\xi}(t_o) = \begin{bmatrix} \xi_1(t_o) \\ \vdots \\ \xi_6(t_o) \end{bmatrix}$$

$$\underline{x}(t_o) = T_M(t_o)\, \underline{\xi}(t_o)$$

$$\underline{X}(t_o) = \underline{X}^*(t_o) + \underline{x}(t_o)$$

Input Quantities Required	Computed Quantities Required
TRFG; HSFG;BSFG SSFG; $M(t_o)$ B_1; B_2; B_3 B_4; B_5 ω; T_L; Θ_i	

Output
DIMFG; H_A; M_A θ_i



3.3.2.2.3 Electromagnetic Sensors Initialization - Block B.3

B. 3.1 Earth Tracker Longitudes

Compute the Greenwich hour angle at T_L

$$G_{HA} = 100.07554 + .98564735\, d_o + 2.9015 \times 10^{-13}\, d_o^2 + \omega' d_1$$

$$\omega' = \omega/(1 + 5.21 \times 10^{-13}\, d_o)$$

$$0 \leq G_{HA} < 360^o$$

where

d_o	=	integral part of $T_L - T_{1950}$
d_1	=	fractional part of $T_L - T_{1950}$
ω	=	Earth's rotation = (.00417807417 deg/sec)(86400 sec/day)
T_{1950}	=	2433282.5 Julian days
T_L	=	Julian Date of $t_o = 0$

The tracker longitudes are then

$$\theta_i = \Theta_i + G_{HA} \qquad i = 1, 2, 3$$

B.3.2 Dimension

INPUT: In this block the overall dimension of the state vector and concomitant matrices and vectors shall be established. This shall be accomplished essentially through the input values assigned to four flags: SSFG, HSFG, TRFG, BSFG. Based on these flags Table B.3 can be used to determine DIMFG. This quantity contains two quantities. The first quantity, say m, designates the overall number of components of the observation vector $\underline{z}_A(t_k)$ (i.e., $\underline{z}$ is mx1. It also determines the total number of rows of $H_A(t_k)$, and the total number of columns of K_A. The second number, say n, defined by DIMFG determines the following dimensions.

i. $\underline{x}_A(t_k)$, $\hat{\underline{x}}_A(t_k)$, and all other state-related vectors are (nx1)

ii. $P_A(t_k)$, $P'_A(t_k)$, $M_A(t_k)$ are (nxn)

iii. $H_A(t_k)$ is (mxn)

iv. $K_A(t_k)$ is (nxm)

In the Navigation Block where the gain matrix is computed, the overall gain matrix $K_A(t_k)$ is not computed explicitly. Instead, gain matrices $K_i(t_k)$ relating to particular instruments are computed which have dimension (nxm^i). In this case,

$$\sum_{i=1}^{k} m^i = m$$

The m^i are determined from the instrument flag according to the following schedule.

SSFG $\neq$ 0; $m^i = 1$

HSFG $\neq$ 0; $m^i = 3$

TRFG = 1,2,3; $m^i = 4$

Table B.3 also presents the structure of the H_A matrix in partitioned form. The equations defining the submatrices naturally are given in the description of Block V.

$$H_A = \begin{bmatrix} H_1 \\ H_2 \\ H_3 \\ H_4 \\ H_5 \end{bmatrix}$$

where H_i represents the observation matrix for the i^{th} instrument.

$$i = \begin{cases} 1 & \text{ground tracking system No. 1} \\ 2 & \text{ground tracking system No. 2} \\ 3 & \text{ground tracking system No. 3} \\ 4 & \text{horizon sensor} \\ 5 & \text{space sextant} \end{cases}$$

The structure of the H_i for each value of DIMFG is described in the table. When H_i for some i is not defined, this should be interpreted as meaning that the H_i will not be of interest in the particular simulation run. (Note: The convention has been adopted that if only one tracker is used, it must be tracker No. 1. Further, two trackers imply No. 1 and No. 2.). The structure of the H_A matrix is then modified by moving the remaining rows up and the remaining columns to the left.

Table B.3 depicts the largest dimension of the H_A with the BSFG $\neq 0$ and all the instruments being used (i.e., TRFG=3, HSFG$\neq$0, and SSFG$\neq$0). Changing the setting of these flags changes the DIMFG and the structure of H_A. The number of rows (m) is established only from the instrument flags, whereas the number of columns (n) is established by the BSFG and the instrument flags. For example,

1. $$\left.\begin{array}{l} \text{BSFG} = 0 \\ \text{TRFG} = 1 \\ \text{HSFG} \neq 0 \\ \text{SSFG} \neq 0 \end{array}\right\} \Longrightarrow H_A \triangleq \begin{bmatrix} H_1 \\ \hline H_4 \\ \hline H_5 \end{bmatrix} \triangleq \begin{bmatrix} H_{1T1} \\ \hline H_H \\ \hline H_S \end{bmatrix}$$

   and the DIMFG is 8 x 6 i.e., m = 8, n = 6.

2. $$\left.\begin{array}{l} \text{BSFG} \neq 0 \\ \text{TRFG} = 1 \\ \text{HSFG} \neq 0 \\ \text{SSFG} \neq 0 \end{array}\right\} \Longrightarrow H_A \triangleq \begin{bmatrix} H_1 \\ \hline H_4 \\ \hline H_5 \end{bmatrix} \triangleq \left[\begin{array}{cccc} H_{1T1} & H_{1T2} & 0 & 0 \\ \hline H_H & 0 & I & 0 \\ \hline H_S & 0 & 0 & I \end{array}\right]$$

   and the DIMFG is 8 x 17 i.e., m = 8, n = 17

Note that when the BSFG = 0 all the instruments called by the flags contain biases as far as the structure of the program is concerned. If it is desirable to simulate biases in only a subset of the instruments called, then this can be accomplished through proper input of the initial bias covariance matrices B_i and is explained in detail in Paragraph 4.0.

The flags also define the structure of the $P_A(t_o)$ and $M_A(t_o)$. When

$$BSFG = 0$$

$$M_A(t_o) = M(t_o)$$

When BIASFG $\neq 0$, the initial structure will depend upon the instrument flags. When all instruments are included, then

$$M_A(t_o) = \begin{bmatrix} M(t_o) & 0 & 0 & 0 & 0 & 0 \\ 0 & B_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & B_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & B_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & B_4 & 0 \\ 0 & 0 & 0 & 0 & 0 & B_5 \end{bmatrix}$$

where

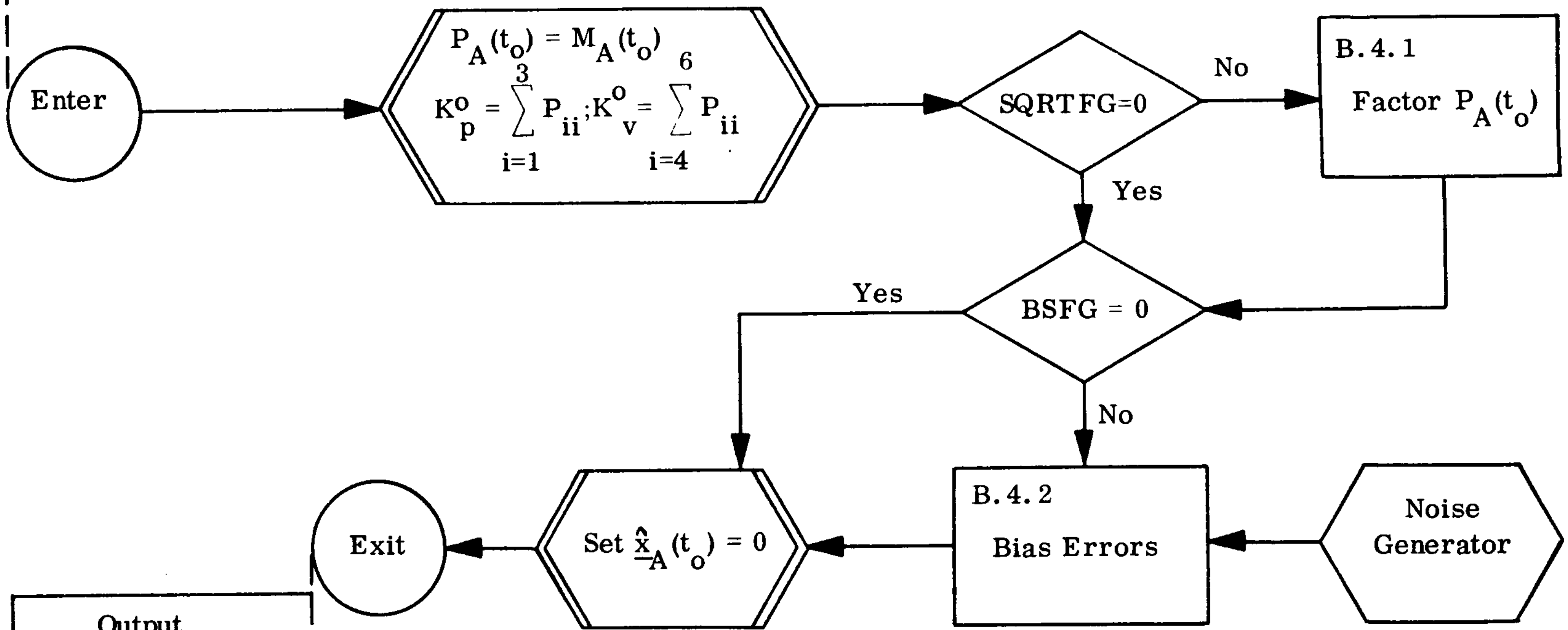
	$M(t_o)$	is (6x6)
	B_1	is (7x7)
	B_2	is (7x7)
	B_3	is (7x7)
	B_4	is (3x3)
	B_5	is (1x1)

The size decreases in an obvious manner depending upon the values of the instrument flags HSFG, SSFG, TRFG.

		State Variable $\underline{x}(t_k)$	Tracker No. 1 Location and Bias Errors $\underline{b}_1$	Tracker No. 2 Location and Bias Errors $\underline{b}_2$	Tracker No. 3 Location and Bias Errors $\underline{b}_3$	Horizon Sensor Bias Errors $\underline{b}_4$	Space Sextant Bias Errors $\underline{b}_5$
Tracker No. 1 $i = 1$	H_1	H_{1T1} (4x6)	H_{1T2} (4x7)	0 (4x7)	0 (4x7)	0 (4x3)	0 (4x1)
Tracker No. 2 $i = 2$	H_2	H_{2T1} (4x6)	0 (4x7)	H_{2T2} (4x7)	0 (4x7)	0 (4x3)	0 (4x1)
Tracker No. 3 $i = 3$	H_3	H_{3T1} (4x6)	0 (4x7)	0 (4x7)	H_{3T2} (4x7)	0 (4x3)	0 (4x1)
Horizon Sensor $i = 4$	H_4	H_H (3x6)	0 (3x7)	0 (3x7)	0 (3x7)	$H_{HB} \triangleq I$ (3x3)	0 (3x1)
Space Sextant $i = 5$	H_5	H_S (1x6)	0 (1x7)	0 (1x7)	0 (1x7)	0 (1x3)	$H_{SB} \triangleq I$ (1x1)

Table B.3 Set-up of DIMFG and Observation Matrix Form

Input Quantities Required	Computed Quantities Required
POFG; BSFG SQRTFG M_{oo} B_{io} $(i=1, 2, \ldots, 5)$	$M_A(t)$



Output
$P_A(t_o)$; $\pi_A(t_o)$ $\underline{b}$; $\hat{\underline{x}}_A(t_o)$; K_p^o; K_v^o

3.3.2.2.4 Navigation Initialization - Block B.4

B.4.1 Factor $P_A(t_o)$

The factorization of $P_A(t_o)$ depends upon two things: the overall dimension of $P_A(t_o)$ and whether or not the matrices to be factored are diagonal.

As defined in B.3., the $M_A(t_o)$ (and, therefore, the $P_A(t_o)$) is composed of submatrices displayed along the principal diagonal $M(t_o)$, $B_1(t_o), \ldots, B_5(t_o)$. Each matrix can be treated separately so the largest matrix that occurs is (6×6).

The input quantities M_{00}, B_{10}, $B_{20}, \ldots, B_{50}$ define the diagonality of the matrices $M(t_o)$, $B_1, \ldots, B_5$.

If $M_{00} = 0$, then $M(t_o)$ is diagonal and the i^{th} diagonal element of the factored matrix is

$$\Pi_{ii}(t_o) = \sqrt{M_{ii}(t_o)}$$

All other elements are zero.

For the other submatrices, the instrument flags must be treated. If the instrument flag (i.e., HSFG) is zero, the corresponding matrix B_i is ignored. When nonzero, the test of the appropriate B_{io} must be made to determine the diagonality (or lack of it) of B_i.

When nondiagonal matrices are involved, the appropriate factorization must be accomplished so that

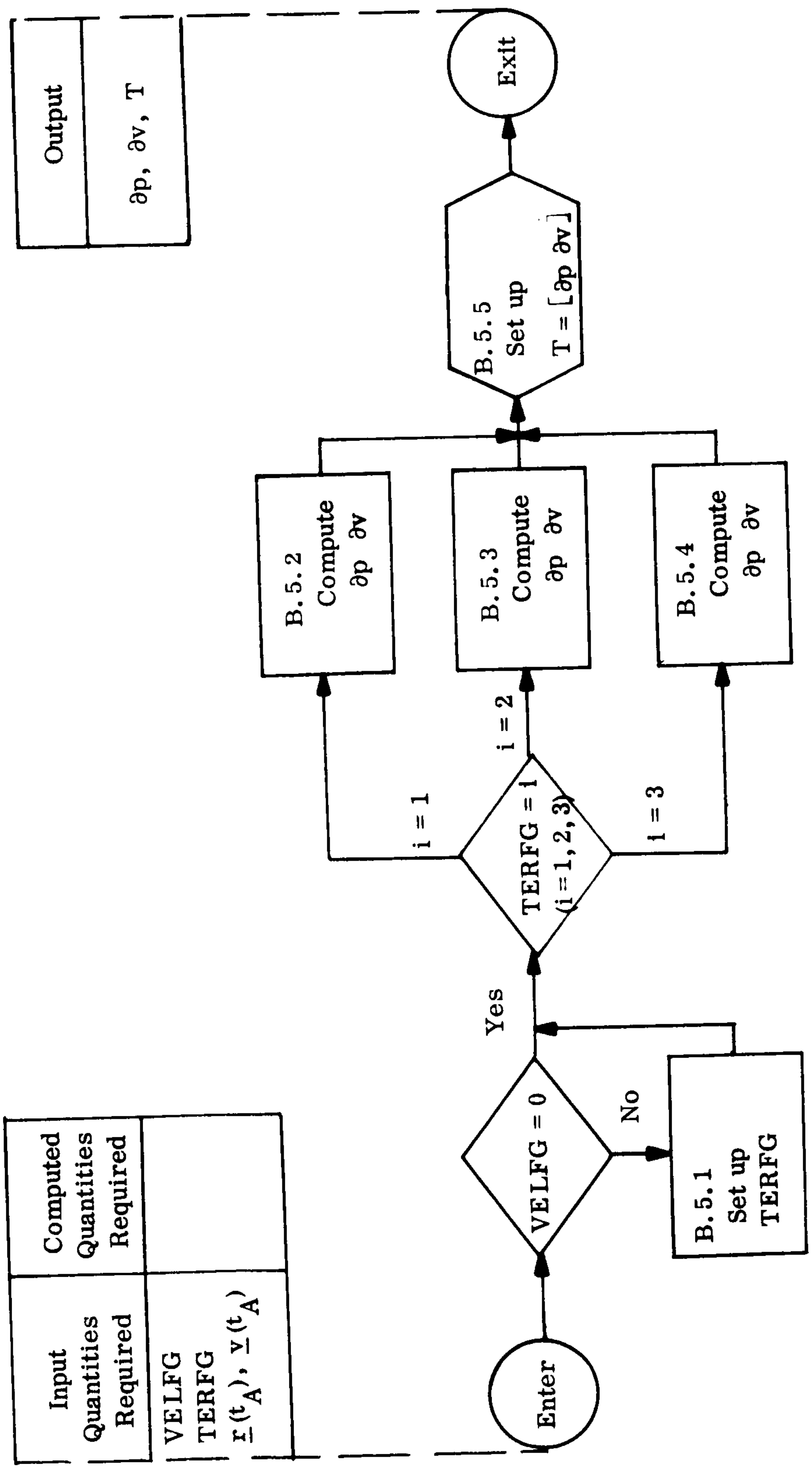
$$B_i = \mathcal{B}_i \, \mathcal{B}_i^T$$

B.4.2 Bias Errors

When there are bias errors, the vector must be generated.

$$\underline{b} = \begin{pmatrix} \underline{b}_1 \\ \vdots \\ \underline{b}_5 \end{pmatrix}$$

where any or all of the subvectors may not appear. The components of each $\underline{b}_i$ are generated from a gaussian number generator using the statistic B_i. When B_i is nondiagonal, the matrix must be factored (3.4.6.2.1) otherwise the diagonal elements of B_i serve as the variances of the components. This vector will not change during the course of the simulation run.



3.3.2.2.5 Guidance Initialization - Block B.5

B.5.1 Set Up of TERFG

If VELFG $\neq 0$, ignore the input TERFG, instead set TERFG according to the following schedule

VELFG = 1,2 → TERFG = 1
VELFG = 3,4 → TERFG = 2
VELFG = 5 → TERFG = 3

B.5.2 TERFG = 1

$$\partial p = I$$

$$\partial v = O$$

$$T = [\partial p \quad \partial v]$$

B.5.3 TERFG = 2

$$\partial p = \begin{bmatrix} \partial p^1 \\ \partial p^2 \\ \partial p^3 \end{bmatrix}$$

$$(\partial p^1)^T = \frac{\underline{r}(t_A^*)}{r(t_A^*)}$$

$$(\partial p^2)^T = \frac{\underline{r}(t_A^*)}{r(t_A^*)} \times \frac{\underline{u}(t_A^*)}{r(t_A^*)}$$

$$(\partial p^3)^T = \underline{0}$$

where

$$r(t_A^*) = [\underline{r}^T(t_A^*)\ \underline{r}(t_A^*)]^{1/2}$$

$$v(t_A^*) = [\underline{v}^T(t_A^*)\ \underline{v}(t_A^*)]^{1/2}$$

$$\partial v \overset{Df}{=} \begin{bmatrix} \partial v^1 \\ \partial v^2 \\ \partial v^3 \end{bmatrix}$$

$$(\partial v^1)^T = \underline{0}$$

$$(\partial v^2)^T = -\frac{\underline{v}(t^*_A)}{v(t^*_A)} \times \frac{\underline{u}(t^*_A)}{v(t^*_A)}$$

$$(\partial v^3)^T = -\frac{\underline{u}(t^*_A)}{v(t^*_A)}$$

where $$\underline{u}(t^*_A) = \frac{\underline{r}(t^*_A) \times \underline{v}(t^*_A)}{\left| \underline{r}(t^*_A) \times \underline{v}(t^*_A) \right|}$$

B.5.4 TERFG = 3

$$\partial p \overset{Df}{=} \begin{bmatrix} \partial p^1 \\ \partial p^2 \\ \partial p^3 \\ \partial p^4 \end{bmatrix}$$

$$(\partial p^1)^T = \frac{\underline{r}(t^*_A)}{r(t^*_A)}$$

$$(\partial p^2)^T = \frac{\underline{r}(t^*_A)}{r(t^*_A)} \times \frac{\underline{u}(t^*_A)}{r(t^*_A)}$$

$$(\partial p^3)^T = \begin{bmatrix} 0 \\ \dfrac{v_z(t^*_A)}{|\underline{r}(t^*_A) \times \underline{v}(t^*_A)|} \\ \dfrac{-v_y(t^*_A)}{|\underline{r}(t^*_A) \times \underline{v}(t^*_A)|} \end{bmatrix}$$

$$(\partial p^4)^T = \begin{bmatrix} \dfrac{-v_z(t^*_A)}{|\underline{r}(t^*_A) \times \underline{v}(t^*_A)|} \\ 0 \\ \dfrac{v_x(t^*_A)}{|\underline{r}(t^*_A) \times \underline{v}(t^*_A)|} \end{bmatrix}$$

where

$$\underline{r}(t^*_A) \overset{Df}{=} \begin{bmatrix} r_x(t^*_A) \\ r_y(t^*_A) \\ r_z(t^*_A) \end{bmatrix} ; \quad \underline{v}(t^*_A) \overset{Df}{=} \begin{bmatrix} v_x(t^*_A) \\ v_y(t^*_A) \\ v_z(t^*_A) \end{bmatrix}$$

$$\partial v \overset{Df}{=} \begin{bmatrix} \partial v^1 \\ \partial v^2 \\ \partial v^3 \\ \partial v^4 \end{bmatrix}$$

$$(\partial v^1)^T = \underline{0}$$

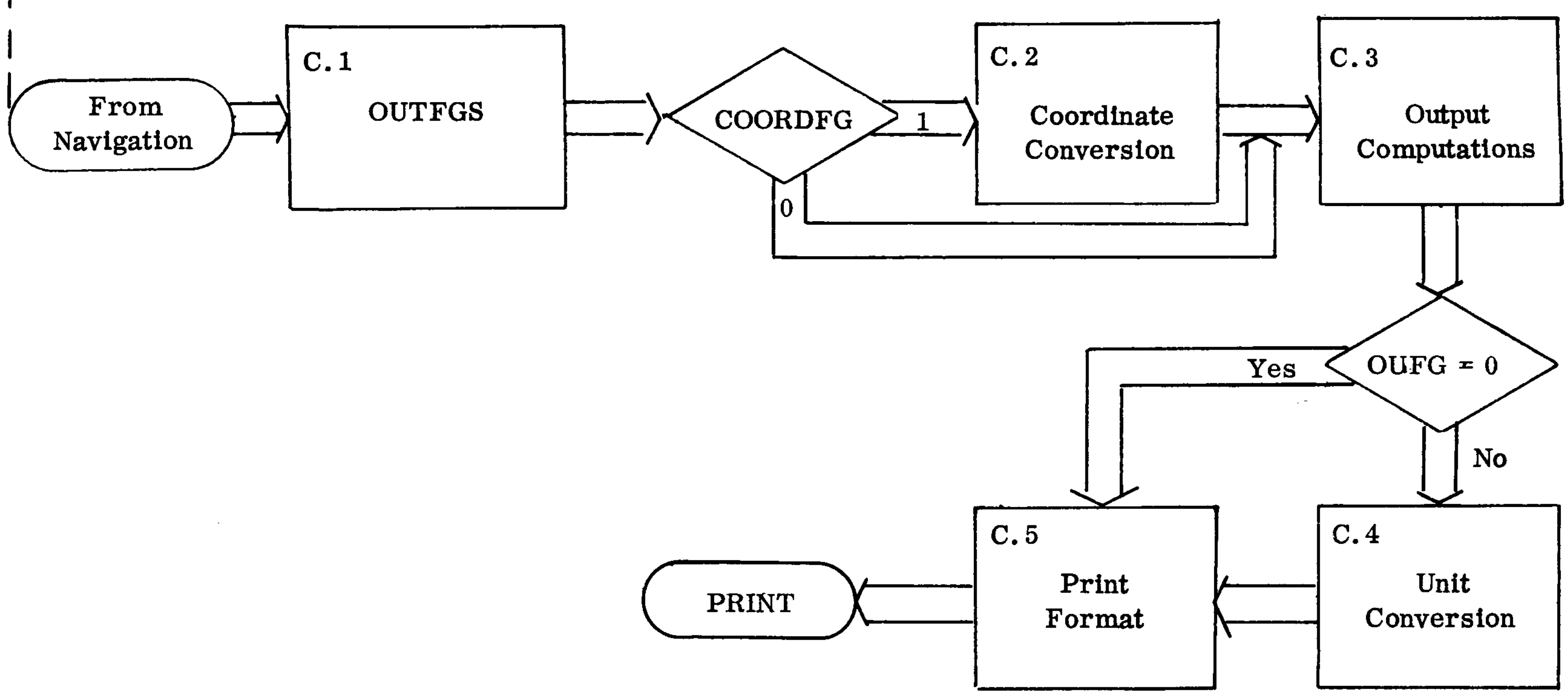
$$(\partial v^2)^T = - \frac{\underline{v}(t_A^*)}{v(t_A^*)} \times \frac{\underline{u}(t_A^*)}{v(t_A^*)}$$

$$(\partial v^3)^T = \begin{bmatrix} \dfrac{-u_x v_x(t_A^*)}{[v(t_A^*)]^2} \\ \dfrac{-r_z(t_A^*)}{|\underline{r}(t_A^*) \times \underline{v}(t_A^*)|} - \dfrac{u_x v_y(t_A^*)}{[v(t_A^*)]^2} \\ \dfrac{r_y(t_A^*)}{|\underline{r}(t_A^*) \times \underline{v}(t_A^*)|} - \dfrac{u_x v_z(t_A^*)}{[v(t_A^*)]^2} \end{bmatrix}$$

$$(\partial v^4)^T = \begin{bmatrix} \dfrac{r_z(t_A^*)}{|\underline{r}(t_A^*) \times \underline{v}(t_A^*)|} - \dfrac{u_y v_x(t_A^*)}{[v(t_A^*)]^2} \\ \dfrac{-u_y v_y(t_A^*)}{[v(t_A^*)]^2} \\ \dfrac{-r_x(t_A^*)}{|\underline{r}(t_A^*) \times \underline{v}(t_A^*)|} - \dfrac{u_y v_z(t_A^*)}{[v(t_A^*)]^2} \end{bmatrix}$$

3.3.3 Output - Block C

Input Quantities Required	Computed Quantities Required
PTRJFG; PSTRFG	PRINFG
PGIDFG; PEMSFG	All the trajectory quantities available
PNAVFG; PLINFG	All the state transition quantities available
COORDFG; OUFG	All the Navigation quantities avail. All the Guidance quantities avail.



3.3.3.1 Level II Flow Chart - Output

3.3.3.2 Detailed Flow Charts and Equations

3.3.3.2.1 Output Print Flags - Block C.1

The output flags are used to specify the desired quantities to be printed. The flags in conjunction with the concomitant program flags establish the overall print format and the availability of quantities to be printed. The desireability of printing at a specific time is established by the print flag (PRINFG). The PRINFG is tied to the observation schedule which is a function of time.

The output format flags are generally tied to the functional blocks in the program. These flags are input as follows:

Trajectory PTRJFG	State Transition PSTRFG	Guidance PGIDFG	Electromagnetic Sensors PEMSFG	Navigation PNAVFG	Linear Approximation PLINFG

Each of the flags above can take on several values. The number zero built into program) is used exclusively to indicate that nothing is printed. The number one (1) indicates the complete print of all the quantities available (see 3.3.3.2.5). The definitions of all the flag designations are given below.

TRAJECTORY BLOCKS

PTRJFG
- = 0 ⟹ No trajectory quantities are printed.
- = 1 ⟹ All the trajectory quantities available are printed. Note that when PRFG = 0, only the nominal quantities are printed.
- = 2 ⟹ Prints trajectory quantities excluding the orbital elements.

STATE TRANSITION BLOCK

PSTRFG
- = 0 ⟹ No print
- = 1 ⟹ Print $\Phi(t_k, t_{k-1})$; $\Phi(t_A, t_k)$; $\Phi(t_k, t_o)$
- = 2 ⟹ Print only $\Phi(t_k, t_{k-1})$
- = 3 ⟹ Print only $\Phi(t_A, t_k)$

GUIDANCE BLOCK

PGIDFG

= 0 ⇛ No print

= 1 ⇛ Print all guidance quantities available

= 2 ⇛ Print $T_E(t_k)$; $\sqrt{T_E \text{ diagonals}}$; $\Lambda(t_k)$; $R_V(t_k)$; trace, volume, eigenvectors, and $\sqrt{\text{eigenvalues}}$ of $V(t_k)$, $D(t_k)$, $N(t_k)$; $\Delta\hat{\underline{V}}(t_k)$ $\Delta\underline{V}(t_k)$; δT_A; $\hat{\underline{x}}'(t_k)$; $M(t_k)$; $P(t_k)$; $P_{11}(t_k)$; $P_{22}(t_k)$; $P_{33}(t_k)$ $P_{44}(t_k)$; $P_{55}(t_k)$; trace, volume, eigenvectors, and $\sqrt{\text{eigenvalues}}$ of $M_1(t_k)$, $M_3(t_k)$, $P_1(t_k)$, $P_3(t_k)$.

Note that the availability of the $P_{ij}(t_k)$ matrices is governed by the DIMFG. $\Delta\hat{\underline{V}}(t_k)$ and $\Delta\underline{V}(t_k)$, and δT_A; t_k are printed whenever a velocity correction is made.

= 3 ⇛ Print $\sqrt{T_E \text{ diagonals}}$; $R_V(t_k)$; trace, volume, eigenvectors, $\sqrt{\text{eigenvalues}}$ of $V(t_k)$, $D(t_k)$, $N(t_k)$; $\Delta\hat{\underline{V}}(t_k)$; $\Delta\underline{V}(t_k)$, δT_A; $\hat{\underline{x}}'(t_k)$; M; P; trace, volume, eigenvectors, $\sqrt{\text{eigenvalues}}$, $M_1(t_k)$, $M_3(t_k)$, $P_1(t_k)$, $P_3(t_k)$.

Note, $\Delta\hat{\underline{V}}(t_k)$, $\Delta\underline{V}(t_k)$ and δT_A; t_k are printed whenever a velocity correction takes place.

= 4 ⇛ Print $\sqrt{T_E \text{ diagonals}}$; $\hat{\underline{x}}'(t_k)$; trace, volume, eigenvectors, $\sqrt{\text{eigenvalues}}$, $M_1(t_k)$, $M_3(t_k)$, $P_1(t_k)$, $P_3(t_k)$.

Note, the following are printed at all the velocity corrections as they occur:

$R_V(t_k)$; trace, volume, eigenvectors, $\sqrt{\text{eigenvalues}}$ of $V(t_k)$ $D(t_k)$, $N(t_k)$; $\Delta\hat{\underline{V}}(t_k)$; $\Delta\underline{V}(t_k)$; δT_A; t_k time of velocity correction.

ELECTROMAGNETIC SENSORS BLOCK

PEMSFG

- = 0 $\Rightarrow$ No print
- = 1 $\Rightarrow$ Print all the electromagnetic quantities available. Note, the observation matrices $H_{iT1}(t_k)$, $H_{iT2}(t_k)$, $i = 1, 2, 3$, $H_S(t_k)$, $H_H(t_k)$ and the measurements $Y_i^*(t_k)$, $Y_i(t_k)$, $i = 1, 2, \ldots, 5$, are printed as a function of the instrument flags and the navigation flags ζ_i, $i = 0, 1, 2, \ldots 5$; i.e., even when an instrument use is indicated in the input but the specific ζ_i indicates that the instrument was not used at the time in question, then the H, Y*, Y of this instrument are not printed. Instead print the following comments:
  When

	$\zeta_0 = 0$	No observations at this time
TRFG $\neq 0$ and	$\zeta_1 = 0$	No observations with first tracker
	$\zeta_2 = 0$	No observations with second tracker
	$\zeta_3 = 0$	No observations with third tracker
HSFG $\neq 0$	$\zeta_4 = 0$	No horizon sensor observations
SSFG $\neq 0$	$\zeta_5 = 0$	No space sextat observations

- = 2 $\Rightarrow$ Print Y_i^*; Y_i $i = 1, 2, \ldots, 5$; STAR; H_{OPT}; BODY. Note, the last three quantities are printed at each observation regardless of the PRINFG.

NAVIGATION BLOCK

PNAVFG

- = 0 $\Rightarrow$ No print
- = 1 $\Rightarrow$ Print all the navigation quantities available
- = 2 $\Rightarrow$ Print $\underline{z}_i(t_k)$, $i = 1, \ldots, 5$; total gain $K(t_k)$; $\hat{\underline{x}}(t_k)$; $\hat{\underline{b}}(t_k)$; $\underline{x}(t_k)$ $\tilde{\underline{x}}(t_k)$; $P'(t_k)$; $P'_{11}(t_k)$; $P'_{22}(t_k)$; $P'_{33}(t_k)$; $P'_{44}(t_k)$; $P'_{55}(t_k)$; trace, volume, eigenvectors, and $\sqrt{\text{eigenvalues}}$ of $P'_1(t_k)$, $P'_3(t_k)$
- = 3 $\Rightarrow$ Print $\underline{z}_i(t_k)$; $\hat{\underline{x}}(t_k)$; $\underline{x}(t_k)$; $\tilde{\underline{x}}(t_k)$; trace, volume, eigenvectors and $\sqrt{\text{eigenvalues}}$ of $P'_1(t_k)$, $P'_3(t_k)$.

LINEAR APPROXIMATION

PLINFG

- = 0 $\Rightarrow$ No print
- = 1 $\Rightarrow$ Print all quantities available

3.3.3.2.2 Coordinate Conversion - Block C.2

The coordinate conversion option is initiated by the COORDFG (input). When the COORDFG = 0, no conversion takes place. When the COORDFG ≠ 0, the quantities delineated below are transformed via an appropriate transformation. At present, provision is made in this block for a single coordinate transformation. When the COORDFG = 1, the output parameters are converted from the internal non-rotating cartesian (i j k) to a rotating cartesian tangent, normal, radial (t n r) coordinate system. The rotating system is defined at each time point as follows:

$$\underline{t} = \underline{n} \times \underline{r} \quad ; \quad \underline{n} = \frac{\underline{R}^*(t_k) \times \underline{V}^*(t_k)}{|\underline{R}^*(t_k) \times \underline{V}^*(t_k)|} \quad ; \quad \underline{r} = \frac{\underline{R}^*(t_k)}{R^*(t_k)}$$

Thus, the transformation matrix between the two systems is given by

$$\mathcal{R}(t_k) = \begin{pmatrix} t_1 & t_2 & t_3 \\ n_1 & n_2 & n_3 \\ r_1 & r_2 & r_3 \end{pmatrix}_{t_k}$$

For print purposes, the converted parameters are printed in the same location, in the print format, as the original parameters.

Nominal

$$\begin{pmatrix} XT^* \\ XN^* \\ XR^* \end{pmatrix}_{t_k} = \mathcal{R}(t_k) \begin{pmatrix} X1^* \\ X2^* \\ X3^* \end{pmatrix}_{t_k} \quad ; \quad \begin{pmatrix} XTD^* \\ XND^* \\ XRD^* \end{pmatrix}_{t_k} = \mathcal{R}(t_k) \begin{pmatrix} X4^* \\ X5^* \\ X6^* \end{pmatrix}$$

Actual

$$\begin{pmatrix} XT \\ XN \\ XR \end{pmatrix}_{t_k} = \mathcal{R}(t_k) \begin{pmatrix} X1 \\ X2 \\ X3 \end{pmatrix}_{t_k} \quad ; \quad \begin{pmatrix} XTD \\ XND \\ XRD \end{pmatrix}_{t_k} = \mathcal{R}(t_k) \begin{pmatrix} X4 \\ X5 \\ X6 \end{pmatrix}_{t_k}$$

Terminal Constraint Matrix T_E

When TERFG = 1,

$$T_E^R(t_k) = \mathcal{R}(t_A)\, T_E(t_k)\, \mathcal{R}^T(t_A)$$

Note, the transformation matrix $\mathcal{R}$ has to be evaluated at t_A.

Velocity Correction Statistics

$$V^R(t_k) = \mathcal{R}(t_k)\, V(t_k)\, \mathcal{R}^T(t_k)$$

$$D^R(t_k) = \mathcal{R}(t_k)\, D(t_k)\, \mathcal{R}^T(t_k)$$

$$N^R(t_k) = \mathcal{R}(t_k)\, N(t_k)\, \mathcal{R}^T(t_k)$$

Orbit Rectification Velocity

$$\Delta\underline{V}^R(t_k) = \mathcal{R}(t_k)\, \Delta\underline{V}(t_k)$$

Velocity Correction

$$\Delta\underline{V}^R(t_k) = \mathcal{R}(t_k)\, \Delta\underline{V}(t_k)$$

$$\Delta\underline{V}^R(t_k) = \mathcal{R}(t_k)\, \Delta\underline{V}(t_k)$$

Extrapolated Estimate

$$\hat{\underline{x}}'^R(t_k) = \begin{pmatrix} \mathcal{R}(t_k) & 0 \\ 0 & \mathcal{R}(t_k) \end{pmatrix} \hat{\underline{x}}'(t_k)$$

Extrapolated Statistics

$$M^R(t_k) = \begin{pmatrix} \mathcal{R}(t_k) & 0 \\ 0 & \mathcal{R}(t_k) \end{pmatrix} M(t_k) \begin{pmatrix} \mathcal{R}^T(t_k) & 0 \\ 0 & \mathcal{R}^T(t_k) \end{pmatrix}$$

$$P^R(t_k) = \begin{pmatrix} \mathcal{R}(t_k) & 0 \\ 0 & \mathcal{R}(t_k) \end{pmatrix} P(t_k) \begin{pmatrix} \mathcal{R}^T(t_k) & 0 \\ 0 & \mathcal{R}^T(t_k) \end{pmatrix}$$

Best Estimate

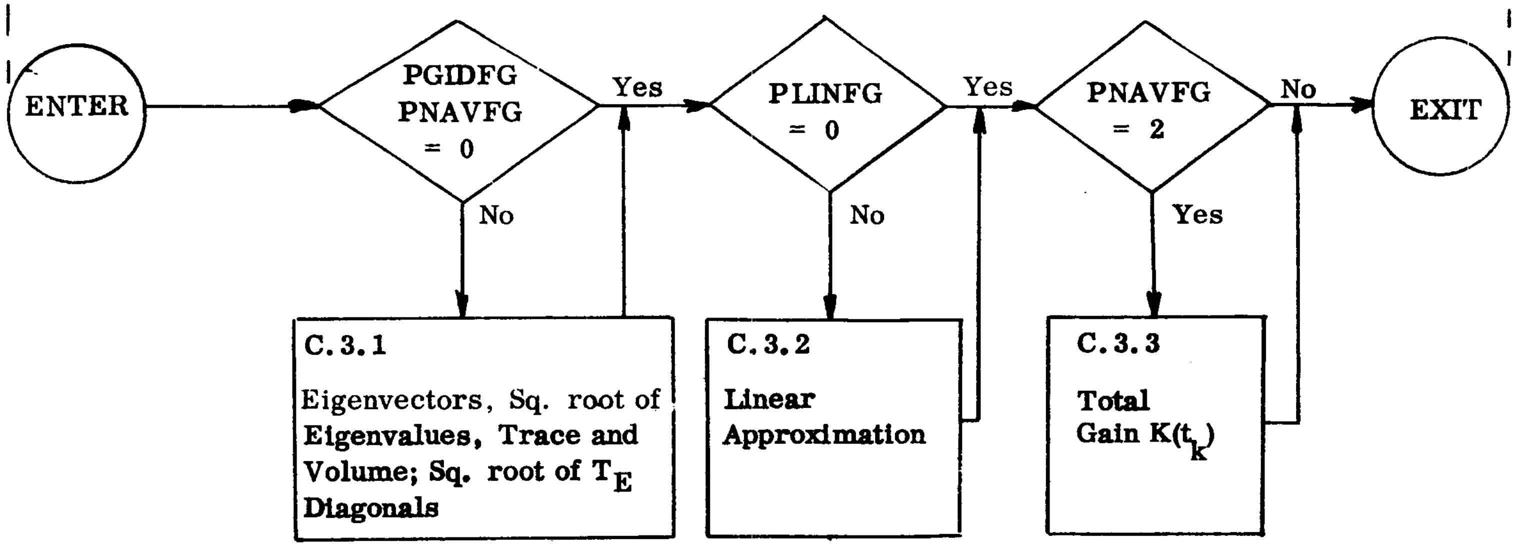
$$\hat{\underline{x}}^R(t_k) = \begin{pmatrix} \mathcal{R}(t_k) & 0 \\ 0 & \mathcal{R}(t_k) \end{pmatrix} \hat{\underline{x}}(t_k)$$

Filtered Statistics

$$P'^R(t_k) = \begin{pmatrix} \mathcal{R}(t_k) & 0 \\ 0 & \mathcal{R}(t_k) \end{pmatrix} P'(t_k) \begin{pmatrix} \mathcal{R}^T(t_k) & 0 \\ 0 & \mathcal{R}^T(t_k) \end{pmatrix}$$

Input Quantities Required	Computed Quantities Required
PGIDFG PNAVFG PLINFG	$V(t_k)$; $D(t_k)$; $N(t_k)$ $M_1(t_k)$; $M_3(t_k)$; $P_1(t_k)$; $P_3(t_k)$ $P'_1(t_k)$; $P'_3(t_k)$; T_E $\underline{X}^*(t_k)$; $\underline{X}(t_k)$; $\hat{x}(t_k)$ $y_i(t_k)$; H_{iT1}; H_5; H_H; $\Phi(t_k, t_o$

Output
$\underline{x}(t_k)$; $\tilde{\underline{x}}(t_k)$ Eigenvectors; Sq. root of Eigenvalues; traces, Volumes; Sq. root of T_E Diagonals; $K(t_k)$; $\Delta y_i(t_k)$; $\Delta\underline{x}(t_k)$



3.3.3.2.3 Output Computations - Block C.3

C.3.1 Square Root of Eigenvalues, Eigenvectors, Trace, Volume, Etc.

Compute the Eigenvalues and Eigenvectors of the matrices V; D; N; M_1; M_3; P_1; P_3; P'_1; P'_3 (subject to availability).

The traces of the matrices considered are given by

$$\sum_{i=1}^{3} a_{ii}$$

The volumes of the matrices are given by

$$\frac{4}{3} \pi \sqrt{\lambda_1} \sqrt{\lambda_2} \sqrt{\lambda_3} \quad \text{where the } \lambda_i\text{'s are the Eigenvalues.}$$

The square roots of the T_E matrix diagonals are simply the $\sqrt{a_{ii}}$.

The magnitude of a vector is given by

$$V = (V_1^2 + V_2^2 + V_3^2)^{1/2}$$

C.3.2 Linear Approximation

Compute:

a. $\underline{x}(t_k) = \underline{X}(t_k) - \underline{X}^*(t_k)$

Note, this computation is done in the internal, as well as converted coordinates when COORDFG $\neq 0$. The internal $\underline{x}(t_k)$ is used in c. below; whereas, the converted $\underline{x}(t_k)$ is output for print purposes and used in d. below when COORDFG $\neq 0$.

b. $\tilde{\underline{x}}(t_k) = \underline{x}(t_k) - \hat{\underline{x}}(t_k)$

Note, this computation is a function of the COORDFG; i.e., when COORDFG = 0, use internal quantities; when COORDF $\neq$ 0, use converted quantities.

c. $\Delta \underline{y}_i(t_k) = \underline{y}_i(t_k) - H_{iT1}\, \underline{x}(t_k) \quad i = 1, 2, 3$

$\Delta \underline{y}_4(t_k) = \underline{y}_4(t_k) - [H_H(t_k) \;\; 0]\, \underline{x}(t_k)$

$\Delta y_5(t_k) = y_5(t_k) - [H_S(t_k) \;\; 0]\, \underline{x}(t_k)$

Note, these computations are governed by the DIMFG and the navigation flags $\zeta_i \quad i = 1, .., 5$.

$\Phi(t_k, t_o)\, \underline{x}(t_o)$

The $\underline{x}(t_k)$, $\underline{x}(t_o)$ employed here is in internal coordinates

d. When COORDFG = 0

$$\Delta\underline{x}(t_k) = \underline{x}(t_k) - \Phi(t_k, t_o)\, \underline{x}(t_o)$$

When COORDFG $\neq$ 0 (= 1)

$$\Delta\underline{x}(t_k) = \underline{x}(t_k) - \mathcal{L}(t_k)\, \Phi(t_k, t_o)\, \underline{x}(t_o)$$

Here $\underline{x}(t_k)$ is the converted one.

C.3.3 Total Gain Matrix $K(t_k)$

The total gain matrix too is governed by the DIMFG and the $\zeta_i \neq 0$ flags, i.e., when a certain $\zeta_i = 0$, then the corresponding $K_i = 0$.

Set $A_i = I - K_i H_i$ $i = 1, \ldots, 5$, then the total gain is given by

$$K(t_k) = [\overbrace{A_5A_4A_3A_2K_1} \quad \overbrace{A_5A_4A_3K_2} \quad \overbrace{A_5A_4K_3} \quad A_5K_4 \quad K_5]$$

3.3.3.2.4 Output Unit Conversion - Block C.4

For output purposes, the program contains a set of conversion factors that enables one to convert from the internal units to desired output units. These conversion factors are:

OUL = output unit of length/internal unit of length

OUT = output unit of time/internal unit of time

The quantities which are converted with their respective multiplicative factors are outlined below. The outline follows the functional block layout of the program output print format 3.6.1.5. Those quantities which are not to be converted are omitted from the outline.

Conic Break Information

(C. CHANGE TIME) · OUT

(POS. VECS.) · OUL ; (VEL. VECS.) · $\left(\frac{\text{OUL}}{\text{OUT}}\right)$

Trajectory

(TIME) · OUT

(POS. VECS.) · OUL ; (VEL. VECS.) · $\frac{\text{OUL}}{\text{OUT}}$; (NOM. A and ACT. A) · OUL

State Transition

Let PHI = $\begin{pmatrix} \text{PHI 1} & \text{PHI 2} \\ \text{PHI 3} & \text{PHI 4} \end{pmatrix}$ then (PHI 2) · OUT ; (PHI 3) · OUT^{-1}

Guidance

Note, when PGIDFG = 1, unit conversion in this block is bypassed and a comment to that effect is printed.

TERFG = 1 (3,3 MATRIX TE) · OUL^2
(SQ.RT.TE DIAG.) · OUL

TERFG = 2 (3,3 MATRIX TE)
(TE 11) · OUL^2
(TE 12, TE 13, TE 21, TE 31) · OUL
(SQ. RT. TE11) · OUL

3,6 LAMBDA

$\Lambda = -[\Lambda_1, I]\ (\Lambda_1) \cdot OUT^{-1}$

(TRACE V, D, N) $\cdot$ $(OUL)^2(OUT)^{-2}$

(3,3 MATRICES V, D, N) $\cdot$ $(OUL)^2(OUT)^{-2}$

(EIGENVALUES SQ.RT V, D, N) $\cdot$ $(OUL)(OUT)^{-1}$

(DEL V VEC., DEL V(HAT) VEC) $\cdot$ $(OUL)(OUT)^{-1}$

(DEL V MAG., DEL V (HAT) MAG) $\cdot$ $(OUL)\ (OUT)^{-1}$

(DELTA) $\cdot$ OUT

1,6 X (HAT) PRM. VEC. = [(1,3) X1 VEC, (1,3) X 2 VEC]

[(1,3) X1 VEC] $\cdot$ OUL ; [(1,3) X2 VEC] $\cdot$ $(OUL)(OUT)^{-1}$

B(HAT)PRM

(1,7) VECS = [(1,4), (1,1), (2,2)]

[(1,4)] $\cdot$ OUL ; [(1,1)] $\cdot$ $(OUL)(OUT)^{-1}$

$$6{,}6\ \text{MATRIX M} = \begin{pmatrix} (3,3)\ M1 & (3,3) M2 \\ (3,3)\ M2^T & (3,3) M3 \end{pmatrix}$$

[(3,3) M1] $\cdot$ $(OUL)^2$

[(3,3) M2 ; (3,3) $M2^T$] $\cdot$ $(OUL)^2(OUT)^{-1}$

[(3,3) M3] $\cdot$ $(OUL)^2(OUT)^{-2}$

$$6{,}6\ \text{MATRIX P} = \begin{pmatrix} (3,3)\ P1 & (3,3)\ P2 \\ (3,3)\ P2^T & (3,3)\ P3 \end{pmatrix}$$

[(3,3) P1] $\cdot$ $(OUL)^2$

[(3,3) P2 ; (3,3) $P2^T$] $\cdot$ $(OUL)^2(OUT)^{-1}$

[(3,3) P3] $\cdot$ $(OUL)^2\ (OUT)^{-2}$

7,7 MATRICES PII

$$\text{PII} = \begin{pmatrix} (4,4)PI1 & (4,1)PI2 & (4,2)PI3 \\ (1,4)PI2^T & (1,1)PI4 & (1,2)PI5 \\ (2,4)PI3^T & (2,1)PI5 & (2,2)PI6 \end{pmatrix}$$

[(4,4) PI1] $\cdot$ $(OUL)^2$ [(1,1) PI_4] $\cdot$ $(OUL)^2(OUT)^{-2}$

[(4,1) PI2 and 1,4 $PI2^T$] $\cdot$ $(OUL)^2(OUT)^{-1}$

[(4,2) PI3 and (2,4) $PI3^T$] $\cdot$ (OUL)

[(1,2) PI5 and (2,1) $PI5^T$] $\cdot$ $(OUL)(OUT)^{-1}$

[EIGENVALUES SQ. RT. M1 and P1] $\cdot$ (OUL)

[EIGENVALUES SQ. RT. M3 and P3] $\cdot$ $(OUL)(OUT)^{-1}$

[TRACE M1 and P1] $\cdot$ $(OUL)^2$

[VOLUME M1 and P1] $\cdot$ $(OUL)^3$

[TRACE M3 and P3] $\cdot$ $(OUL)^2(OUT)^{-2}$

[VOLUME M3 and P3] $\cdot$ $(OUL)^3(OUT)^{-3}$

<u>Electromagnetic Sensors</u> Note, when PEMSFG = 1, unit conversion in this block is bypassed and a comment to that effect is printed.

[NOM. and ACT. MEAS. MAG. RHOS] $\cdot$ (OUL)

[NOM and ACT MEAS MAG. RHODS] $\cdot$ $(OUL)(OUT)^{-1}$

<u>Navigation</u> Note, when PNAVFG = 1, unit conversion in this block is bypassed and a comment to that effect is printed.

[OBS. SM Z(1), Z(2), Z(3) RHO] $\cdot$ (OUL)

[OBS. SM Z(1), Z(2), Z(3) RHOD] $\cdot$ $(OUL)(OUT)^{-1}$

1,6 X (HAT) VEC. = [(1,3) X1 VEC, (1,3) X2 VEC]

[(1,3) X1 VEC] $\cdot$ (OUL) ; [(1,3) X2 VEC] $\cdot$ $(OUL)(OUT)^{-1}$

1,6 DEL X VEC. = [(1,3) DEL X 1, (1,3) DEL X 2]

[(1,3) DEL X1] $\cdot$ (OUL) ; [(1,3) DEL X 2] $\cdot$ $(OUL)(OUT)^{-1}$

1,6 X (WIGL) VEC = [(1,3) X (WIGL) 1, (1,3) X (WIGL) 2]

[(1,3) X(WIGL) 1] $\cdot$ (OUL) ; [(1,3) X(WIGL) 2] $\cdot$ $(OUL)(OUT)^{-1}$

B(HAT) Same conversion as B(HAT)PRM

6,6 P(PRM) Same conversion as 6,6 P

7,7 MATRICES PII(PRM) Same conversion as 7,7 MATRICES PII

[EIGENVALUES SQ. RT. P1(PRM)] · (OUL)

[TRACE P1 (PRM)] $(OUL)^2$; [VOLUME P1 (PRM)] · $(OUL)^3$

[EIGENVALUES SQ. RT. P3 (PRM)] · $(OUL)(OUT)^{-1}$

[TRACE P3 (PRM)] · $(OUL)^2(OUT)^{-2}$; [VOLUME P3 (PRM)] · $(OUL)^3(OUT)^{-3}$

Linear Approximation

[SM DEL Y(1), Y(2), Y(3) RHO] · (OUL)

[SM DEL Y(1), Y(2), Y(3) RHOD] · $(OUL)(OUT)^{-1}$

1,6 SM DEL X = [(1,3) SM DELX1, (1,3) SM DELX2]

[(1,3) SM DELX1] · (OUL) ; [(1,3) SM DELX2] · $(OUL)(OUT)^{-1}$

3.3.3.2.5 Print Format - Block C.5

The print format is composed of two sections, the input print format, 3.3.1.1 and the output print format, C.5.1. The input format is printed only once at the beginning of each simulation run. The input format contains all the information pertinent to the run whether it was part of the input or build in. The output format, however, is governed by the output print flags and can fluctuate in size. The two formats are closely tied to the functional blocks of the program. For an outline of the input format see 3.3.1.1. The most general output format follows.

C. 5. 1 OUTPUT PRINT FORMAT

CONIC BREAK INFORMATION (Note, printed only at occurrence.)

CONIC CHANGE TIME	OLD CENTER	NEW CENTER ......
OLD NOM POS VEC	......	......
OLD NOM VEL VEC	......	......
OLD ACT. POS VEC	......	......
OLD ACT. VEL VEC	......	......
NEW NOM POS VEC	......	......
NEW NOM VEL VEC	......	......
NEW ACT. POS VEC	......	......
NEW ACT. VEL VEC	......	......

TRAJECTORY

TIME	CENTRFG		COORDFG		
NOM POS VEC	......	......	......	POS MAG	
NOM VEL VEC	......	......	......	VEL MAG	
NOM ECC	NOM A		NOM N OR N1		NOM E OR F ......
ACT. POS VEC	......	......	......	POS MAG	
ACT. VEL VEC	......	......	......	VEL MAG	
ACT ECC	ACT. A		ACT. N OR N1		ACT E OR F ......

STATE TRANSITION

6,6 MATRIX PHI (TK, TK-1)

......
......
....

6,6 MATRIX PHI (TA, TK)

......
......
....

6,6 MATRIX PHI (TK, TO)

......
......
....

GUIDANCE

3,3 OR 4,4 MATRIX TE

......
......
.... (......)

SQ RT OF TE MATRIX DIAGONALS

....

3,3 MATRIX LAMBDA (TK)

......
......
....

VELOCITY CORRECTION STATISTICS

RV ...... TRACE V(TK) ...... TRACE D(TK) ...... TRACE N(TK) ......

3,3 MATRIX V(TK)

......
......
....

```
1,3 EIGENVECTORS V(TK).....     ......      ......     EIGENVALUES V(TK) SQ RT = ......
                   .....        ......      ......                               ......
                   .....        ......      ......                               ......

3,3 MATRIX D(TK)
......
......
......        ......      ......

1,3 EIGENVECTORS D(TK) = ......   ......    ......     EIGENVALUES D(TK) SQ RT = ......
                         ......   ......    ......                               ......
                         ......   ......    ......                               ......

3,3 MATRIX N(TK)
......
......
......        ......      ......

1,3 EIGENVECTORS N(TK)   ......   ......    ......     EIGENVALUES N(TK) SQ RT   ......
                         ......   ......    ......                               ......
                         ......   ......    ......                               ......

ORBIT RECT DEL V VEC     ......   ......    ......     DELTA ......
VEL CORRECTION   DEL V(HAT) VEC ......      ......     ......
   DELTA ......  DEL V VEC      ......      ......     ......
                 NOISE VEC      ......      ......     ......
```

Note, Orbit Rectification and Velocity Correction Information always printed at occurrence.

```
EXTRAPOLATED ESTIMATE
1,6X(HAT) PRM VEC        ......   ......    ......   ......   ......   ......
B(HAT) PRM
   1,7 VEC  ......       ......   ......    ......   ......   ......   ......
   1,7 VEC  ......       ......   ......    ......   ......   ......   ......
   1,7 VEC  ......       ......   ......    ......   ......   ......   ......
   1,3 VEC  ......       ......   ......
   1,1 VEC  ......
```

EXTRAPOLATED STATISTICS

6, 6 MATRIX M

. . . . . .
. . . . . .
. . . .

N, N MATRIX PIA AND PA

$$PA = \begin{pmatrix} 6x6\ P & X & X & X & X & X \\ X & 7x7\ P11 & X & X & X & X \\ X & X & 7x7\ P22 & X & X & X \\ X & X & X & 7x7\ P33 & X & X \\ X & X & X & X & 3x3\ P44 & X \\ X & X & X & X & X & 1x1\ P55 \end{pmatrix}$$

Note, the total dimensions of B(HAT) PRM and PIA or PA depend upon the DIMFG. The PIA matrix is printed only when the SQRTFG $\neq$ 0.

1,3 EIGENVECTORS M1(TK) . . . . EIGENVALUES M1(TK) SQ RT . . . . . .
. . . .
. . . .

TRACE M1(TK) . . . . . . VOLUME M1(TK) . . . . . .

1,3 EIGENVECTORS M3(TK) . . . . EIGENVALUES M3(TK) SQ RT . . . . . .
. . . .
. . . .

TRACE M3(TK) . . . . . . VOLUME M3(TK) . . . . . .

1,3 EIGENVECTORS P1(TK) . . . . EIGENVALUES P1(TK) SQ RT . . . . . .
. . . .
. . . .

TRACE P1(TK) . . . . . . VOLUME P1(TK) . . . . . .

1,3 EIGENVECTORS P3(TK)	......	......	......	EIGENVALUES P3(TK) SQ RT ......
	......	......	......	......
	......	......	......	......
TRACE P3(TK)	......	VOLUME P3(TK)		

Note $P = \begin{pmatrix} P1 & P2 \\ P2^T & P3 \end{pmatrix}$ $M = \begin{pmatrix} M1 & M2 \\ M2^T & M3 \end{pmatrix}$

<u>ELECTROMAGNETIC SENSORS</u>

EPHEMERIS INFORMATION

TK ...... (double-precision Julian days) CENTRFG ......

PLNT 0 POS VEC	......	......	......
PLNT 0 VEL VEC	......	......	......
PLNT 1 POS VEC	......	......	......
PLNT 1 VEL VEC	......	......	......
.	.	.	.
.	.	.	.
PLNT 6 POS VEC	......	......	......
PLNT 6 VEL VEC	......	......	......

TRACKER INFORMATION

SM R1T VEC	......	......	......
SM R2T VEC	......	......	......
SM R3T VEC	......	......	......
NOM RHO 1 VEC	......	......	......
NOM RHO 2 VEC	......	......	......
NOM RHO 3 VEC	......	......	......

NOM RHOD1 VEC	......	......	......	
NOM RHOD2 VEC	......	......	......	
NOM RHOD3 VEC	......	......	......	
ACT. RHO1 VEC	......	......	......	
ACT. RHO2 VEC	......	......	......	
ACT. RHO3 VEC	......	......	......	
ACT. RHOD1 VEC	......	......	......	
ACT. RHOD2 VEC	......	......	......	
ACT. RHOD3 VEC	......	......	......	
	MAG RHO	MAG RHOD	PSI	ETA
NOM MEAS Y(1)*	......	......	......	......
Y(2)*	......	......	......	......
Y(3)*	......	......	......	......
ACT MEAS Y(1)	......	......	......	......
Y(2)	......	......	......	......
Y(3)	......	......	......	......

INSTRUMENT COVARIANCES

4,4 MATRIX R(1)

......

......

....

4,4 MATRIX R(2)

......

......

....

4,4 MATRIX R(3)

......

......

....

OBSERVATION MATRICES

4,6 MATRIX H(1T1)

......
......
....

4,7 MATRIX H(1T2)

......
......
....

4,7 MATRIX H2T2)

......
......
....

4,6 MATRIX H3T1)

......
......
......

4,7 MATRIX H3T2)

......
......
......

HORIZON SENSOR

NOM ANG MEAS Y(4)* ....
ACT ANG MEAS Y(4) ....
3,3 MATRIX INST COV R(4) ....
....
....

3,3 MATRIX H(H) ....
....
....

SPACE SEXTANT

NOM MEAS Y(5)* ..... ACT MEAS Y(5) ...... INST COV R(5) ...... BODY ......

1, 3 MATRIX H(S) ....

1, 3 MATRIX H(OPT) ....

STAR ...... UN.ST. VEC. ....

Note, print of instrument covariances, observation matrices, and measurements subject to the DIMFG and observation flags ζ_i $i = 1, \ldots, 5$.

NAVIGATION

SM Y(1)	......	......	......	......
SM Y(2)	......	......	......	......
SM Y(3)	......	......	......	......
SM Y(4)	......	......	......	
SM Y(5)	......			
NOISE V(1)	......	......	......	......
NOISE V(2)	......	......	......	......
NOISE V(3)	......	......	......	......
NOISE V(4)	......	......	......	
NOISE V(5)	......			
OBS SMZ(1)	......	......	......	......
OBS SMZ(2)	......	......	......	......
OBS SMZ(3)	......	......	......	......
OBS SMZ(4)	......	......	......	
OBS SMZ(5)	......			

GAIN MATRICES

31, 4 MATRIX K(1)

31, 4 MATRIX K(2)

31, 4 MATRIX K(3)

31, 3 MATRIX K(4)

31, 1 MATRIX K(5)

31, 16 MATRIX TOTAL GAIN K

FILTERED ESTIMATE

1, 6 X(HAT) VEC	......	......	......	......	......	......	
1, 6 DEL X VEC	......	......	......	......	......	......	
1, 6 X(WIGL) VEC	......	......	......	......	......	......	
B(HAT) 1, 7 VEC	......	......	......	......	......	......	......
1, 7 VEC	......	......	......	......	......	......	......
1, 7 VEC	......	......	......	......	......	......	......
1, 3 VEC	......	......	......				
1, 1 VEC	......						

FILTERED STATISTICS

N, N MATRIX PIA(PRM) AND PA(PRM)

$$
\text{PIA(PRM)} = \begin{pmatrix}
6,6\ \text{PI} & X & X & X & X & X \\
X & 7,7\ \text{PI11} & X & X & X & X \\
X & X & 7,7\ \text{PI22} & X & X & X \\
X & X & X & 7,7\ \text{PI33} & X & X \\
X & X & X & X & 3,3\ \text{PI44} & X \\
X & X & X & X & X & 1,1\ \text{PI55}
\end{pmatrix}
$$

$$\text{PA(PRM)} = \begin{pmatrix} \text{6, 6 P} & \text{X} & \text{X} & \text{X} & \text{X} & \text{X} \\ \text{X} & \text{7, 7 P11} & \text{X} & \text{X} & \text{X} & \text{X} \\ \text{X} & \text{X} & \text{7, 7 P22} & \text{X} & \text{X} & \text{X} \\ \text{X} & \text{X} & \text{X} & \text{7, 7 P33} & \text{X} & \text{X} \\ \text{X} & \text{X} & \text{X} & \text{X} & \text{3, 3 P44} & \text{X} \\ \text{X} & \text{X} & \text{X} & \text{X} & \text{X} & \text{1, 1 P55} \end{pmatrix}$$

1,3 EIGENVECTORS P1(PRM)		......	......	......	EIGENVALUES P1(PRM) SQ RT	......
		......	......	......		......
		......	......	......		......
TRACE P1(PRM)	......	VOLUME P1(PRM)				
1,3 EIGENVECTORS P3(PRM)		......	......	......	EIGENVALUES P3(PRM) SQ RT	......
		......	......	......		......
		......	......	......		......
TRACE P3(PRM)	......	VOLUME P3(PRM)				

<u>LINEAR APPROXIMATION</u>

1, 4 SM DEL Y(1)	......	......	......	......		
1, 4 SM DEL Y(2)	......	......	......	......		
1, 4 SM DEL Y(3)	......	......	......	......		
1, 3 SM DEL Y(4)	......	......	......			
1, 1 SM DEL Y(5)	......					
1, 6 SM DEL X	......	......	......	......	......	......

3.3.3.3 Purpose of the Functional Block Outputs

FUNCTIONAL BLOCK	QUANTITY	WHERE REQUIRED	
General Initialization	$\underline{X}(t_o)$	Actual	Output
	$P'_A(t_o)$		Output
	$\underline{b}$	Navigation	Output
	$\hat{\underline{x}}_A(t_o)$	Guidance	Output
	$\pi'_A(t_o)$	Guidance	Output
	$M_A(t_o)$	Guidance	Output
	DIMFG	Guidance	
Nominal	$\underline{X}^*(t_k)$	Actual Electromagnetic Sensors	Output
	$R^*(t_k)$	State Transition	
	t_k	Electromagnetic Sensors	Output
	$\tau^*(t_k)$		Output
	$E^*(t_k)$	State Transition	Output
	$F^*(t_k)$	State Transition	Output
	$n^*(t_o)$	State Transition	Output
	$n_1^*(t_o)$	State Transition	Output
	$e^*(t_o)$	State Transition	Output
	Δt_k	State Transition, Actual	
	FILTFG	Actual	
State Transition	CENTRFG	Nominal	
	$\Phi(t_k, t_o)$		Output
	$\Phi(t_k, t_{k-1})$	Guidance	Output
	$\Phi(t_A, t_k)$	Guidance	Output
	$\Phi(t_o^m, t_o^o)$		
	$\Phi(t_o^{m-1}, t_o^m)$		
	STMFG	Nominal	
	M	Actual	

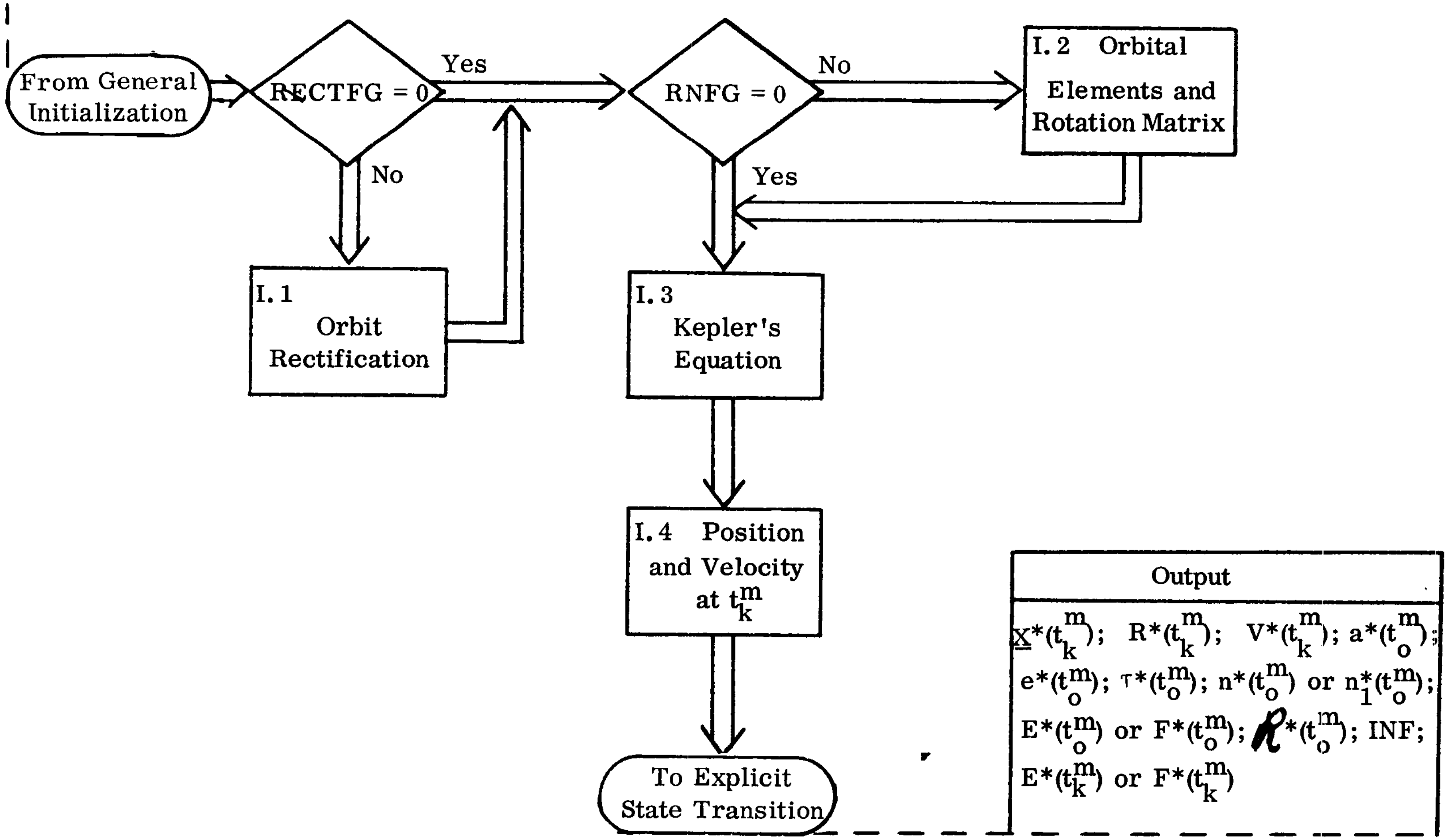
FUNCTIONAL BLOCK	QUANTITY	WHERE REQUIRED	
Actual	$\underline{X}(t_o)$	Electromagnetic Sensors	Output
	FILTFG	Actual	
	CENTRFG	Nominal	
	CONFG	Nominal	
Guidance	CORRFG	Actual	
	$M_A(t_k)$	Guidance	Output
	$P_A(t_k)$	Navigation	Output
	$\hat{\underline{x}}'(t_k)$	Navigation	Output
	T_E		Output
	$\hat{\underline{x}}^R(t_k)$	Nominal	Output
	$\Lambda(t_k)$		Output
	C_τ		Output
	$\overline{d^2}(t_k)$		Output
	$\overline{\Delta\hat{v}^2}(t_k)$		Output
	$\overline{n^2}(t_k)$		Output
	R_V		Output
	$\Delta\hat{\underline{V}}(t_k)$		Output
	$\Delta\underline{V}(t_k)$	Actual	Output
	δT	Nominal	Output
	$P'_A(t_k)$	Guidance	Output
	$\hat{\underline{x}}_A(t_k)$	Nominal	Output
Electromagnetic Sensors $i = 1, 2, \ldots, 5$	$H_i(t_k)$	Navigation	Output
	$\underline{Y}^*_i(t_k)$	Navigation	Output
	$\underline{Y}_i(t_k)$	Navigation	Output
	ζ_o	Navigation	
	ζ_i	Navigation	
	$R_i(t_k)$	Navigation	

FUNCTIONAL BLOCK	QUANTITY	WHERE REQUIRED	
Navigation	$\underline{z}_i(t_k)$		Output
	$\hat{\underline{x}}_A(t_k)$	Guidance	Output
	$K_i(t_k)$		Output
	$P'_A(t_k)$	Guidance	Output
	$\pi'_A(t_k)$	Guidance	Output
	RECTFG	Nominal, Guidance	

3.4 BASIC COMPUTATIONAL BLOCKS

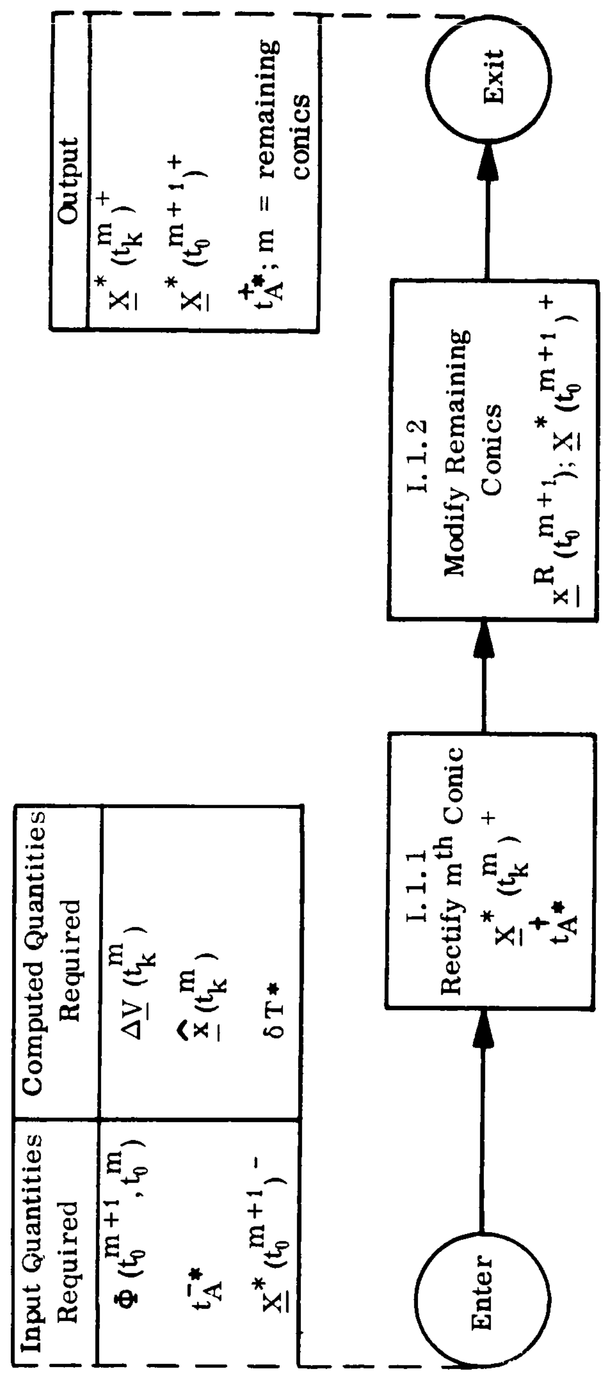
3.4.1 Two-Body Interplanetary Nominal - Block I

Input Quantities Required	Computed Quantities Required
$\underline{X}^*(t_o^m) = \begin{bmatrix} \underline{R}^*(t_o^m) \\ \underline{V}^*(t_o^m) \end{bmatrix}$; t_o^m m = 0, 1, . . . 9; μ_ℓ ℓ = 0, 1, . . . 6; MF	RECTFG; δt_A^*; $\hat{\underline{x}}(t_{k-1}^m)$; $\Delta \underline{V}^*(t_{k-1}^m)$ $\Phi(t_o^{m+1}, t_o^m)$ m = 0, 1, . . . 9; m; RNFG; ; $\underline{X}^*(t_{k-1}^m)$; t_k; Δt_k; $E^*(t_{k-1})$ or $F^*(t_{k-1})$



3.4.1.1 Level II Flow Chart - Two-Body Interplanetary Nominal

3.4.1.2 Detailed Flow Charts and Equations



3.4.1.2.1 Orbit Rectification - Block I. 1

I. 6. 1 Rectify mth Conic

$$\underline{x}^R(t_k^m) = \hat{\underline{x}}(t_k^m) + \begin{bmatrix} 0 \\ \Delta\underline{V}(t_k^m) \end{bmatrix}$$

$$\underline{X}^*(t_k^m)^+ = \underline{X}^*(t_k^m)^- + \underline{x}^R(t_k^m) \qquad \underline{X}^*(t_k^m) = \begin{bmatrix} \underline{R}^*(t_k^m) \\ \underline{V}^*(t_k^m) \end{bmatrix}$$

$$(t_A^+)^* = (t_A^-)^* + \delta T^*$$

where + designates after rectification

- designates before rectification

and t_A^+ affects only the ΔT in the last conic.

Set $\Delta\underline{V}^R(t_k^m) \equiv 0$ and $\hat{x}(t_k^m) = \binom{0}{\Delta\underline{V}R}$ for the next cycle.

I. 6. 2 Modify the Remaining Conics

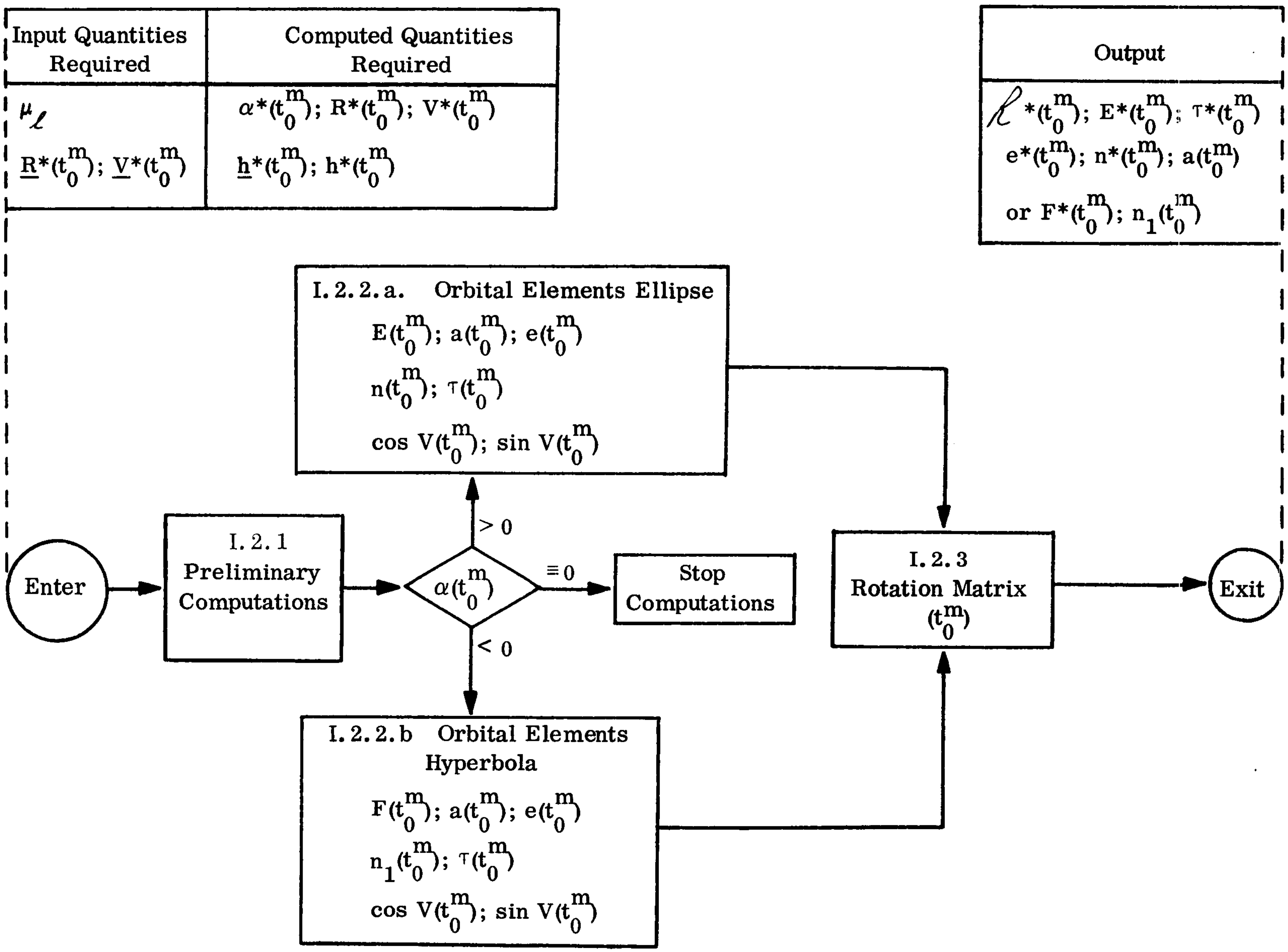
$$\underline{x}^R(t_o^{m+1}) = \Phi(t_o^{m+1}, t_k^m)\,\underline{x}^R(t_k^m)$$

where

$$\Phi(t_o^{m+1}, t_k^m) = \Phi(t_o^{m+1}, t_o^m)\,\Phi^{-1}(t_k^m, t_o^m)$$

$$\underline{X}^*(t_o^{m+1})^+ = \underline{X}^*(t_o^{m+1})^- + \underline{x}^R(t_o^{m+1})$$

After computing all the remaining conditions $\underline{x}^R(t_o^m)$ and initial states $\underline{X}^*(t_o^m)^+$, a set of new total state transition matrices $\Phi(t_o^{m+1}, t_o^m)$ for the remaining conics are computed. This set replaces the "old" set for future use. The ΔT^m's in all the conics are unchanged except the last one which is affected by t_A^*.



3.4.1.2.2 Orbital Elements and Rotation Matrix - Block I.2

I. 2. 1 **Preliminary Computations**

Compute:

Time in m^{th} conic $\Delta T^m = t_o^{m+1} - t_o^m$

Position magnitude $R^*(t_o^m) = [\underline{R}^*(t_o^m) \cdot \underline{R}^*(t_o^m)]^{1/2}$

Velocity magnitude $V^*(t_o^m) = [\underline{V}^*(t_o^m) \cdot \underline{V}^*(t_o^m)]^{1/2}$

Angular momentum vector $\underline{h}^*(t_o^m) = \underline{R}^*(t_o^m) \times \underline{V}^*(t_o^m)$

Angular momentum magnitude $h^*(t_o^m) = [\underline{h}^*(t_o^m) \cdot \underline{h}^*(t_o^m)]^{1/2}$

From the energy establish type of conic

$$\alpha^*(t_o^m) = \frac{2}{R^*(t_o^m)} - \frac{\underline{V}^*(t_o^m) \cdot \underline{V}^*(t_o^m)}{\mu_\ell}$$

$$\alpha^*(t_o^m) \begin{cases} > 0 \rightarrow \text{ellipse} \\ < 0 \rightarrow \text{hyperbola} \\ = 0 \rightarrow \text{parabola (stop computations).} \end{cases}$$

where t_o^m is the initial time of the m^{th} conic, and the * designates the nominal values. Since the equations are the same in both the actual and nominal cases, the * superscript will be retained only in the input and output.

I. 2. 2 **Nominal Orbital Elements with Respect to Orbital Plane**

a. $\alpha\ (t_o^m) > 0 \Rightarrow$ elliptic case

$$a(t_o^m) = \frac{1}{\alpha(t_o^m)}$$

$$e(t_o^m) = \{(1 - \frac{R(t_o^m)}{a(t_o^m)})^2 + \frac{[\underline{R}(t_o^m) \cdot \underline{V}(t_o^m)]^2}{\mu_1\ a(t_o^m)}\}^{1/2}$$

If $e(t_o^m) \leq 1 \times 10^{-6}$, set $e(t_o^m) = 0$; $E(t_o^m) = 0$ and bypass 4.1.1.6, the solution of Kepler's equation; instead compute $E(t_k^m) = E(t_{k-1}^m) + n\Delta t_k$.

$$E(t_o^m) = \begin{cases} \cos^{-1} \dfrac{1}{e(t_o^m)} \left[1 - \dfrac{R(t_o^m)}{a(t_o^m)}\right] \\ \sin^{-1} \dfrac{1}{e(t_o^m)} \left\{ \dfrac{\underline{R}(t_o^m) \cdot \underline{V}(t_o^m)}{[\mu_1 \, a(t_o^m)]^{1/2}} \right\} \end{cases}$$

$$n(t_o^m) = \sqrt{\frac{\mu_\ell}{a^3(t_o^m)}}$$

$$\cos \nu(t_o^m) = \frac{\cos E(t_o^m) - e(t_o^m)}{1 - e(t_o^m) \cos E(t_o^m)}$$

$$\sin \nu(t_o^m) = \frac{\sqrt{1 - e^2(t_o^m)} \, \sin E(t_o^m)}{1 - e(t_o^m) \cos E(t_o^m)}$$

$$\tau(t_o^m) = \frac{1}{n(t_o^m)} [E(t_o^m) - e \sin E(t_o^m)]$$

b. $\alpha(t_o^m) < 0 \rightarrow$ hyperbolic case

To simplify the notation, from here on, the functional dependence upon time; i.e., $f(t_o^m)$ will be omitted where it is obvious.

$$a = \frac{1}{\alpha}$$

$$e = \{(1 - \frac{R}{a})^2 + \frac{(\underline{R} \cdot \underline{V})^2}{\mu_1 a}\}^{1/2}$$

$$\cosh F = \frac{1}{e} (1 - \frac{R}{a})$$

$$\sinh F = \frac{1}{e} (\frac{\underline{R} \cdot \underline{V}}{\sqrt{-\mu_\ell a}})$$

compute τ from $n_1 \tau = -F + e \sinh F$

where $n_1 = \sqrt{\dfrac{\mu_2}{-a^3}}$

$$\cos \nu = \frac{\cosh F - e}{1 - e \cosh F}$$

$$\sin \nu = \frac{-\sqrt{e^2 - 1} \sinh F}{1 - e \cosh F}$$

I. 2. 3 Rotation Matrix Recomputed at the Start of Each New Conic

$$\mathcal{Q} = \begin{pmatrix} \frac{X}{R} & \eta_x & \frac{h_x}{h} \\ \frac{Y}{R} & \eta_y & \frac{h_y}{h} \\ \frac{Z}{R} & \eta_z & \frac{h_z}{h} \end{pmatrix} \begin{pmatrix} \cos \nu & \sin \nu & 0 \\ -\sin \nu & \cos \nu & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

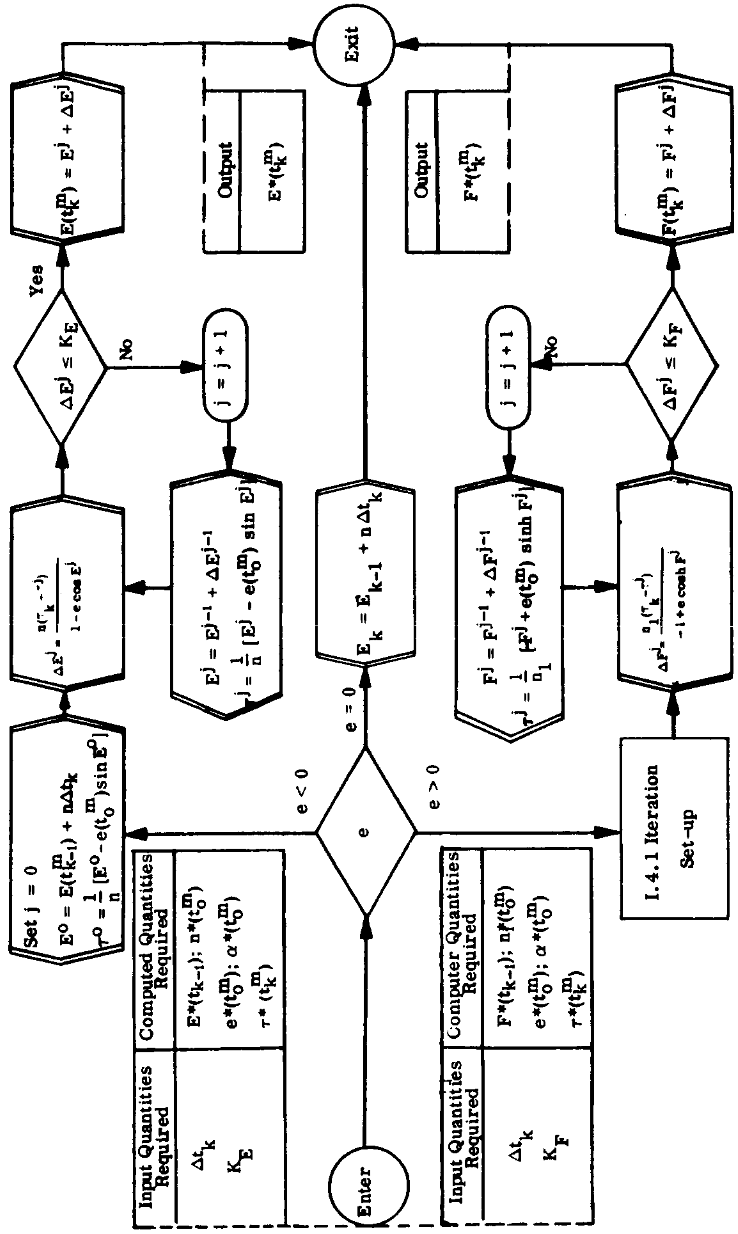
$$\underline{\eta} = \frac{\underline{h}}{h} \times \frac{\underline{R}}{R}$$

i. e. $\eta_x = \frac{1}{hR} (x_3 h_y - x_2 h_z)$

$\eta_y = \frac{1}{hR} (x_1 h_z - x_3 h_x)$

$\eta_z = \frac{1}{hR} (x_2 h_x - x_1 h_y)$

$\mathcal{Q}$ is computed only once at the beginning of each conic and is used each Δt up to the next conic when it is recomputed. In the nominal case, $\mathcal{Q}$ is also recomputed after an orbit rectification.



3.4.1.2.3 Kepler's Equation - Block I.3

I. 4. 1 Iteration Set-up

Set $j = 0$

Compute $M_k = M_{k-1} + n_1 \Delta t_k$

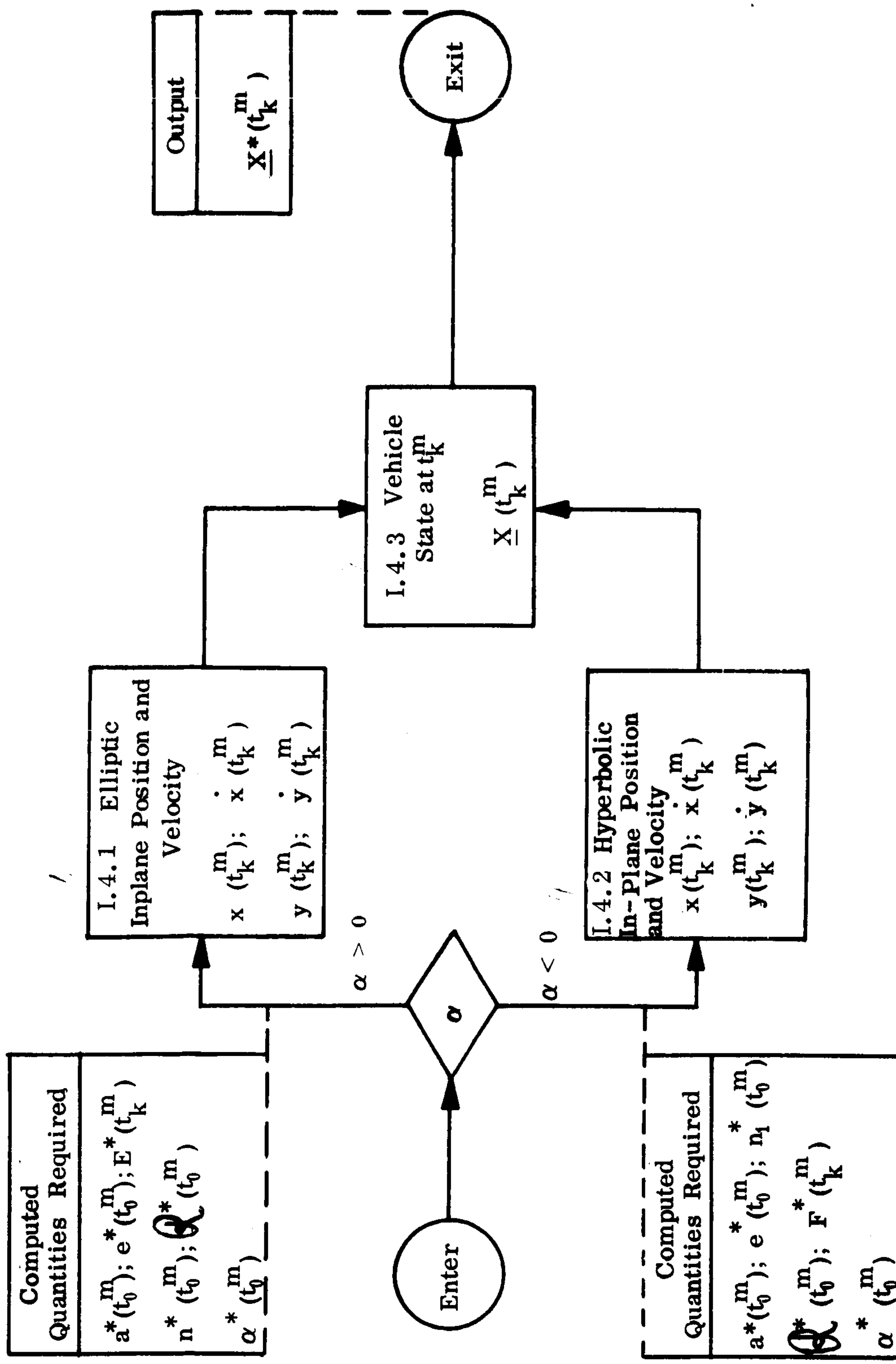
$$M_{k-1} = -F_{k-1} + e \sinh F_{k-1}$$

Let

$$F^o = \begin{cases} F_{k-1} & \text{if } n_1 \Delta t_k \leq 5 \\ \left.\begin{array}{ll} (6M_k)^{1/2} & \text{if } |M_k| \leq 1 \\ \sinh^{-1}\left(\frac{M_k}{e}\right) & \text{if } |M_k| \geq 1 \end{array}\right\} & \text{if } n_1 \Delta t_k > 5 \end{cases}$$

Compute

$$\tau^o = \frac{1}{n_1} \left[-F^o + e \sinh F^o \right]$$



3.4.1.2.4 Position and Velocity at t_k^m - Block I.4

I.4.1 Elliptic, in-plane position and velocity

$$x(t_k^m) = a(t_o^m)[\cos E(t_k^m) - e(t_o^m)]$$

$$y(t_k^m) = a(t_o^m)\sqrt{1 - e^2(t_o^m)}\,\sin E(t_k^m)$$

$$\dot{x}(t_k^m) = a(t_o^m)\,n(t_o^m)\left[\frac{-\sin E(t_k^m)}{1 - e(t_o^m)\cos E(t_k^m)}\right]$$

$$\dot{y}(t_k^m) = a(t_o^m)\,n(t_o^m)\sqrt{1 - e^2(t_o^m)}\left[\frac{\cos E(t_k^m)}{1 - e(t_o^m)\cos E(t_k^m)}\right]$$

note!

$$r(t_k^m) = \sqrt{x^2(t_k^m) + y^2(t_k^m)} = a(t_o^m)[1 - e(t_o^m)\cos E(t_k^m)]$$

I.4.2 Hyperbolic, in-plane position and velocity

$$x(t_k^m) = a(t_o^m)[\cosh F(t_k^m) - e(t_o^m)]$$

$$y(t_k^m) = -\,a(t_o^m)\sqrt{e^2(t_o^m) - 1}\,\sinh F(t_k^m)$$

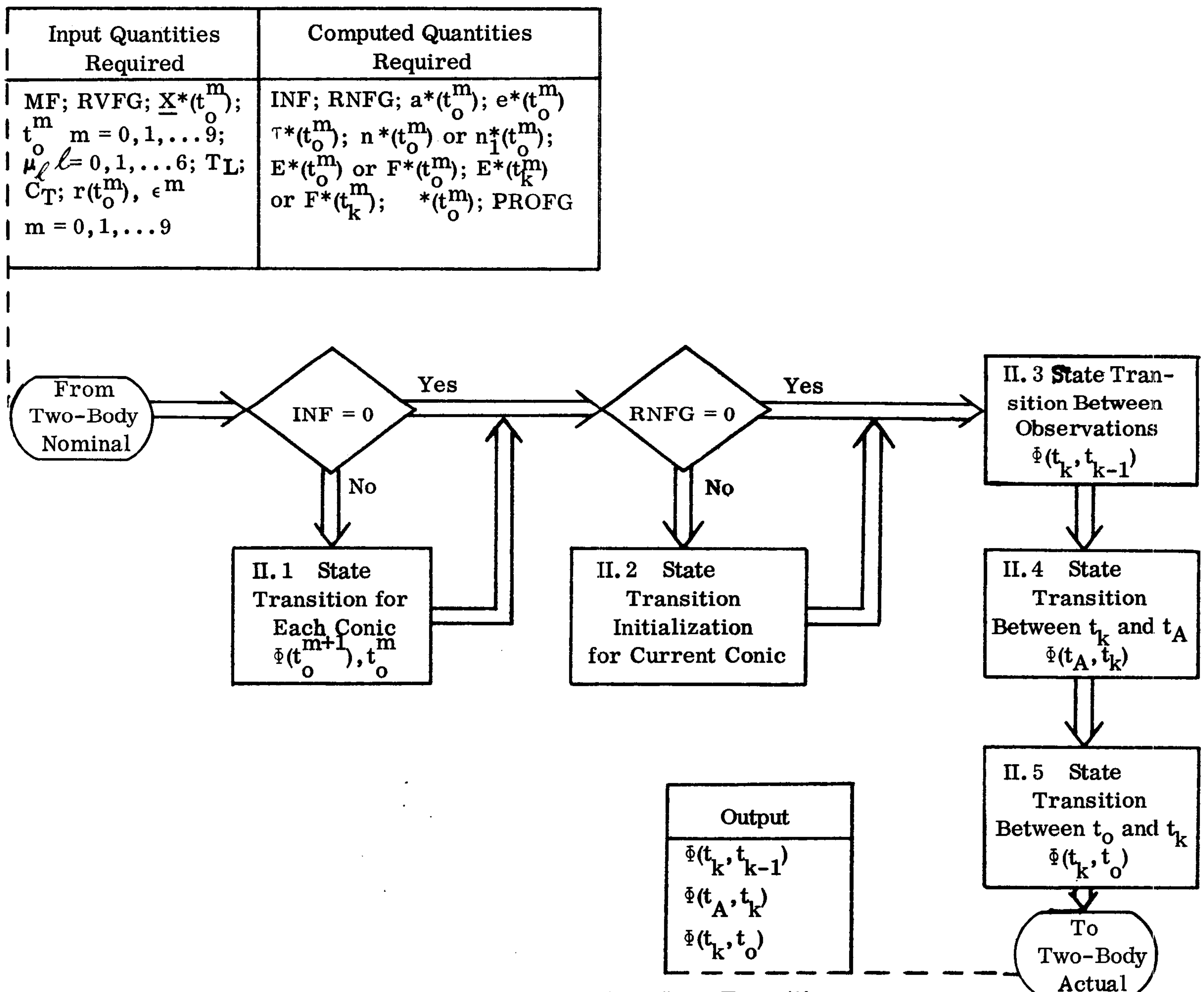
$$\dot{x}(t_k^m) = a(t_o^m)\,n_1(t_o^m)\left[\frac{-\sinh F(t_k^m)}{1 - e(t_o^m)\cosh F(t_k^m)}\right]$$

$$\dot{y}(t_k^m) = a(t_o^m)\,n_1(t_o^m)\sqrt{e^2(t_o^m) - 1}\left[\frac{\cosh F(t_k^m)}{1 - e(t_o^m)\cosh F(t_k^m)}\right]$$

I.4.3 Vehicle State at t_k^m

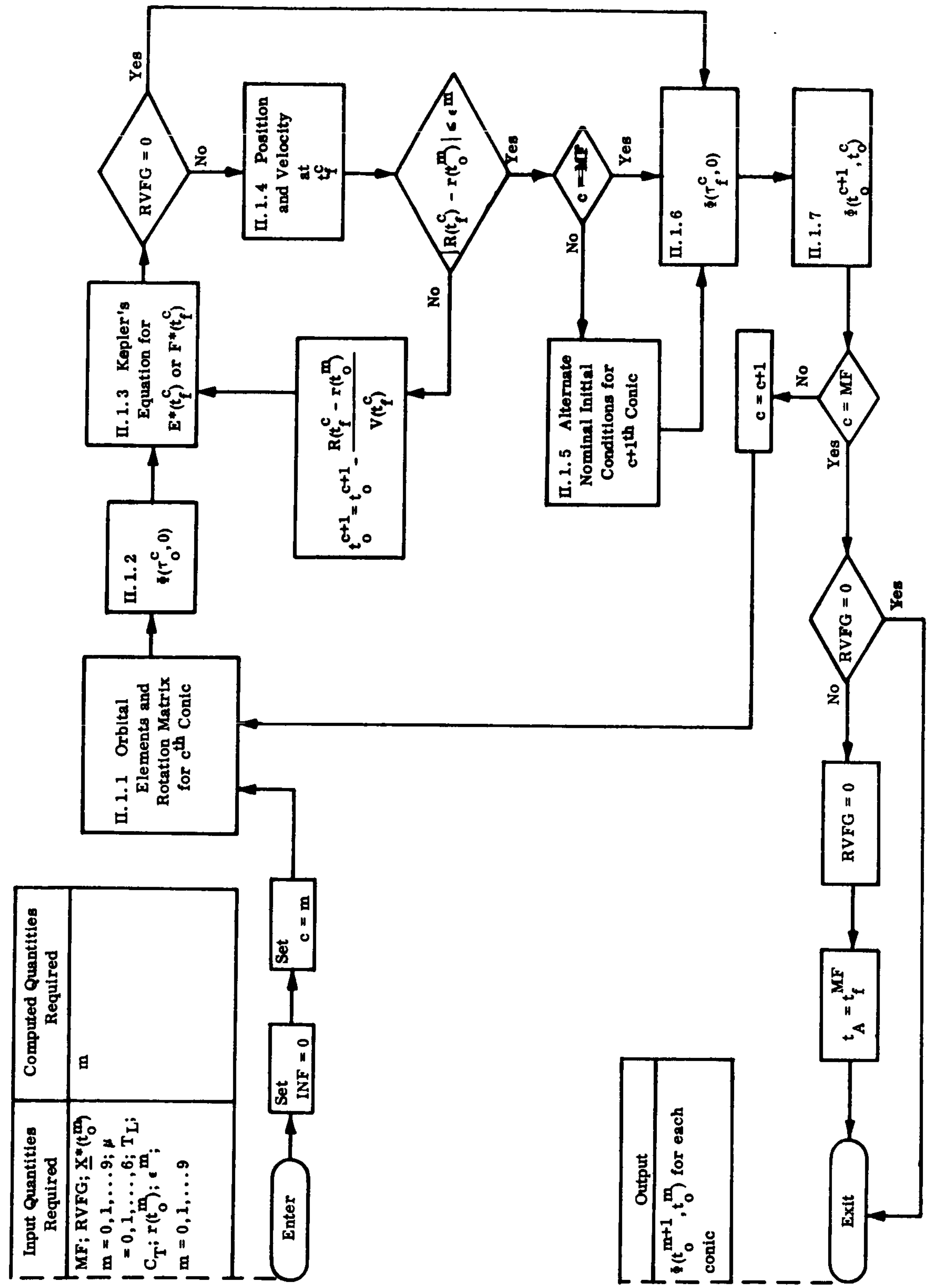
$$\underline{X}(t_k^m) = \begin{bmatrix} \underline{R}(t_k^m) \\ \underline{V}(t_k^m) \end{bmatrix} = \begin{bmatrix} \mathcal{Q}(t_o^m) & 0 \\ 0 & \mathcal{Q}(t_o^m) \end{bmatrix} \begin{bmatrix} x(t_k^m) \\ y(t_k^m) \\ 0 \\ \dot{x}(t_k^m) \\ \dot{y}(t_k^m) \\ 0 \end{bmatrix}$$

3.4.2 Explicit State Transition - Block II



3.4.2.1 Level II Flow Chart - Explicit State Transition

3.4.2.2 Detailed Flow Charts and Equations



3.4.2.2.1 State Transition for Each Conic - Block II.1

II. 1. 1 Orbital Elements and Rotation Matrix for c^{th} Conic

Employ equations in I. 2 with m = c to compute orbital elements and rotation matrix.

II. 1. 2 In-Plane State Transition Matrix

II. 1. 2a With $E_k \triangleq E^*(t_o^c)$ or $F^*(t_o^c)$, employ equations in II. 1. 2b to obtain

$$\Phi(\tau_o^c, 0) \ \{ = \Phi(\tau_k, 0)\}.$$

II. 1. 2b State Transition Equations

Employing E_k, compute

$$\Phi(\tau_k, 0) = \begin{pmatrix} \Phi_1 & \Phi_2 \\ \Phi_3 & \Phi_4 \end{pmatrix}$$

where the submatrices Φ_i i = 1, 2, 3, 4 are given by

$$\Phi_1 = \begin{pmatrix} \Phi_{11} & \Phi_{12} & 0 \\ \Phi_{21} & \Phi_{22} & 0 \\ 0 & 0 & \Phi_{33} \end{pmatrix} \quad \Phi_2 = \begin{pmatrix} \Phi_{14} & \Phi_{15} & 0 \\ \Phi_{24} & \Phi_{25} & 0 \\ 0 & 0 & \Phi_{36} \end{pmatrix}$$

$$\Phi_3 = \begin{pmatrix} \Phi_{41} & \Phi_{42} & 0 \\ \Phi_{51} & \Phi_{52} & 0 \\ 0 & 0 & \Phi_{63} \end{pmatrix} \quad \Phi_4 = \begin{pmatrix} \Phi_{44} & \Phi_{45} & 0 \\ \Phi_{54} & \Phi_{55} & 0 \\ 0 & 0 & \Phi_{66} \end{pmatrix}$$

When $\alpha > 0$, let $\sin E(t_k) = S_k$; $\cos E(t_k) = C_k$; $E(t_k) = E_k$

or when $\alpha < 0$, set $\sinh F(t_k) = S_k$; $\cosh F(t_k) = C_k$; $F(t_k) = E_k$

$$\Phi_{11} = \frac{1}{(1-e)^2 (1-e\,C_k)} \left[C_k^2 (1+e-e^2) + C_k (2+e+2e^2-e^3) - 2 - 5e + 2e^2 + 3 \frac{a}{|a|} E_k S_k \right]$$

$$\Phi_{12} = \frac{\frac{a}{|a|}\sqrt{|1-e^2|}}{(1-e)(1-e\,C_k)} S_k (1 - C_k)$$

$$\Phi_{21} = \frac{\frac{a}{|a|}\sqrt{|1-e^2|}}{(1-e)^2(1-e\,C_k)} \left[S_k C_k (1+e) + S_k (2-e) - 3\,E_k C_k\right]$$

$$\Phi_{22} = \frac{1}{(1-e)(1-e\,C_k)} \left[C_k^{\,2} + C_k(-1-2e+e^2) + 1\right]$$

$$\Phi_{33} = \frac{(C_k - e)}{(1-e)}$$

$$\Phi_{14} = \frac{|(1-e)|}{n(1-e\,C_k)} S_k \left[-C_k(1+e) + 2\right]$$

$$\Phi_{15} = \frac{\frac{a}{|a|}\sqrt{|1-e^2|}}{(1-e)\,n(1-e\,C_k)} \left[C_k^{\,2}(2-e) + 2\,C_k(1+e) - 4 - e + 3\frac{a}{|a|} E_k S_k\right]$$

$$\Phi_{24} = \frac{\frac{a}{|a|}\sqrt{|1-e^2|}}{n(1-e\,C_k)} \left[1 - C_k\right]^2$$

$$\Phi_{25} = \frac{\frac{a}{|a|}}{n(1-e\,C_k)} \left[S_k C_k (2+e+e^2) + 2\,S_k - 3(1+e)\,E_k C_k\right]$$

$$\Phi_{36} = \frac{S_k|(1-e)|}{n}$$

$$\Phi_{41} = \frac{n}{(1-e)^2(1-e\,C_k)^3} \left[S_k C_k^{\,2}(e+e^2-e^3) + S_k C_k(-2-5e+2e^2)\right.$$
$$\left. + S_k(1+e+3e^2-e^3) + 3\,E_k(C_k - e)\right]$$

$$\Phi_{42} = \frac{\frac{a}{|a|} n\sqrt{|1-e^2|}}{(1-e)(1-e\,C_k)^3} \left[e\,C_k^{\,3} - 2\,C_k^{\,2} + C_k + 1 - e\right]$$

$$\Phi_{51} = \frac{\frac{a}{|a|} n\sqrt{|1-e^2|}}{(1-e)^2(1-e\,C_k)^3} \left[-C_k^{\,3}(e+e^2) + C_k^{\,2}(2+5e) - C_k(1+e)\right.$$
$$\left. - 1 - 3e + e^2 + 3\frac{a}{|a|} E_k S_k\right]$$

$$\Phi_{52} = \frac{n\,S_k}{(1-e)\,(1-e\,C_k)^3}\,[e\,C_k^2 - 2\,C_k + 1 + e - e^2]$$

$$\Phi_{63} = \frac{-n\,S_k}{(1-e)\,(1-e\,C_k)}$$

$$\Phi_{44} = \frac{1-e}{(1-e\,C_k)^3}\,[C_k^3(e+e^2) - 2\,C_k^2(1+e) + 2\,C_k + 1 - e]$$

$$\Phi_{45} = \frac{\frac{a}{|a|}\sqrt{|1-e^2|}}{(1-e)\,(1-e\,C_k)^3}\,[S_kC_k^2(2e-e^2) - S_kC_k(4+e) + S_k(1+e)^2 + 3\,E_k(C_k-e)]$$

$$\Phi_{54} = \frac{\frac{a}{|a|}S_k\sqrt{|1-e^2|}}{(1-e\,C_k)^3}\,[e\,C_k^2 - 2\,C_k + 2 - e]$$

$$\Phi_{55} = \frac{1}{(1-e\,C_k)^3}\,[-C_k^3(2e+e^2+e^3) + C_k^2(4+5e+5e^2) - C_k(1+3e) - 2 - 3e - e^2 + 3\frac{a}{|a|}(1+3)\,E_kS_k]$$

$$\Phi_{66} = \frac{(1-e)\,C_k}{(1-e\,C_k)}$$

II.1.3 Kepler's Equation for $E^*(t_f^c)$ or $F^*(t_f^c)$

Using the orbital elements for the c^{th} conic and $\Delta t_k = \Delta T^c = t_o^{c+1} - t_o^c$, employ the equations of 3.4.1.2.3 to obtain $E^*(t_f^c)$ or $F^*(t_f^c)$.

II.1.4 Position and Velocity at t_f^c

Using the orbital elements for the c^{th} conic and $E^*(t_f^c)$ or $F^*(t_f^c)$ for E_k, employ the equations of 3.4.1.2.4 to obtain $\underline{X}^*(t_f^c)$.

II.1.5 Alternate Nominal Initial Conditions for $c+1^{th}$ Conic

$$\underline{X}^*(t_o^{c+1}) = \underline{X}^*(t_f^c) + \underline{X}_{c+1,c}(T_o^{c+1})$$

where $\underline{X}_{c+1,c}$ is the state vector (position and velocity) of the central body for conic c with respect to the central body for conic c+1 obtained from the ephemeris tape at time

$$T_o^{c+1} = T_L + C_T(t_o^c).$$

II.1.6

With $E_k \triangleq E(t_f^c)$ or $F(t_f^c)$, employ equations in II.1.2b to obtain

$$\Phi(\tau_f^c, 0) = \{\Phi(\tau_k, 0)\}$$

II.1.7 State Transition Matrix for c^{th} Conic and $\Phi(t_o^{c+1}, t_o^c)$

With $E_k \triangleq E(t_f^c)$ or $F(t_f^c)$ and $E_{k-1} \triangleq E(t_o^c)$ or $F(t_o^c)$, and using the orbital elements and rotation matrix for the c^{th} conic, employ the equations of II.1.7b to obtain

$$\Phi(t_o^{c+1}, t_o^c) \triangleq \{\Phi(t_k, t_{k-1})\} \triangleq \Phi(t_f^c, t_o^c)$$

where

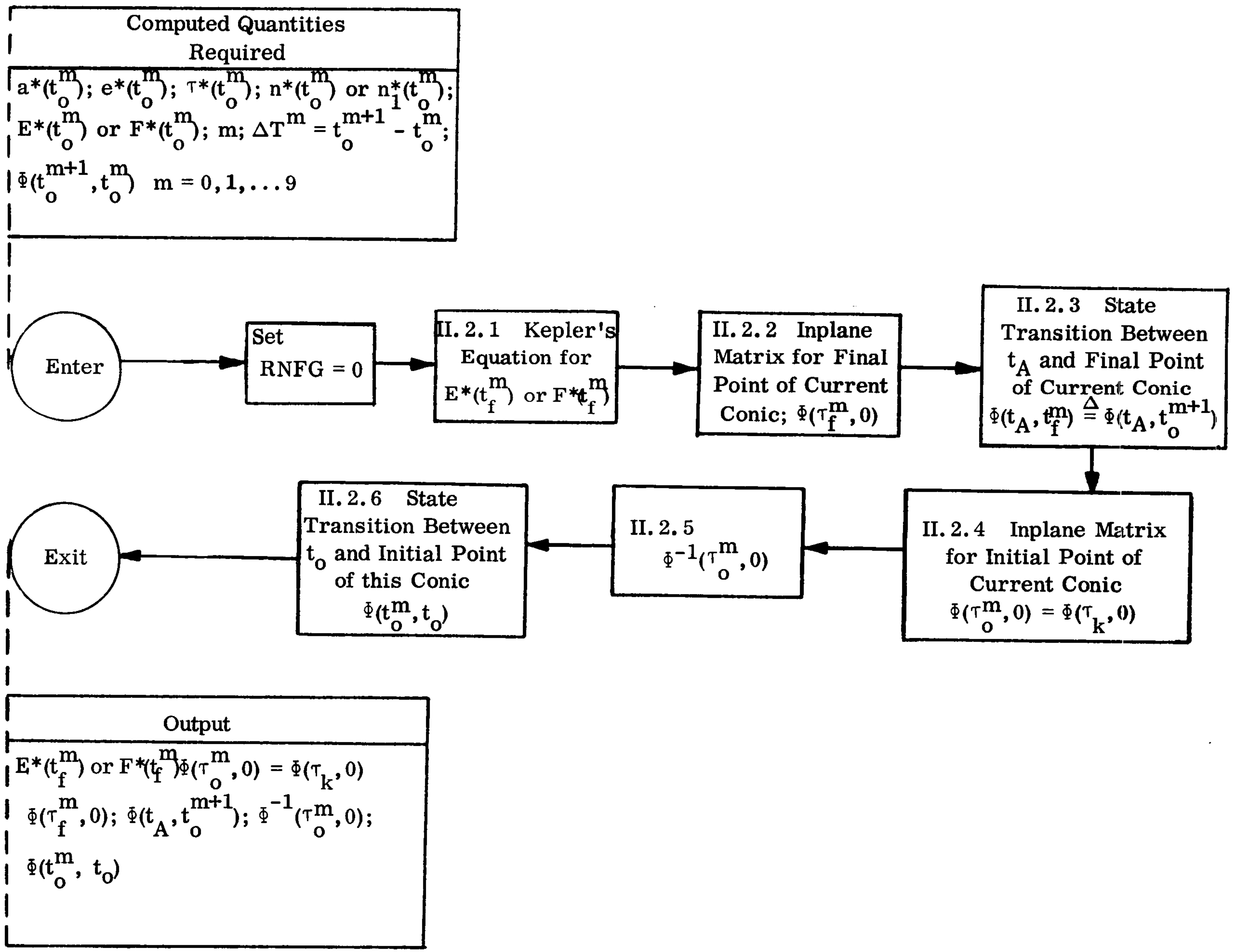
$$\Phi(t_k, t_{k-1}) = \mathcal{R}(t_o^c)\, \Phi(\tau_k, 0)\, \Phi^{-1}(\tau_{k-1}, 0)\, \mathcal{R}^T(t_o^c)$$

and the inverse is obtained as follows; if

$$\Phi(\tau_k, 0) = \begin{bmatrix} \Phi_1 & \Phi_2 \\ \Phi_3 & \Phi_4 \end{bmatrix}$$

then

$$\Phi^{-1}(\tau_k, 0) = \begin{bmatrix} \Phi_4^T & -\Phi_2^T \\ -\Phi_3^T & \Phi_1^T \end{bmatrix}$$



3.4.2.2.2 State Transition Initialization for Current Conic - Block II.2

II.2.1 Kepler's Equation for $E^*(t_f^m)$ or $F^*(t_f^m)$

Using the orbital elements of the current conic, and $\Delta t_k = \Delta T^m = t_o^{m+1} - t_o^m$, employ the equations of 3.4.1.2.3 to obtain $E^*(t_f^m)$ or $F^*(t_f^m)$.

II.2.2 In-plane Matrix for Final Point of Current Conic

With $E_k \triangleq E(t_f^m)$ or $F(t_f^m)$, employ equations in II.1.2b to obtain

$$\Phi(\tau_f^m, 0) \{ = \Phi(\tau_k, 0) \}$$

II.2.3 State Transition $\Phi(t_A, t_f^m) = \Phi(t_A, t_o^{m+1})$

If $m = MF$, $\Phi(t_A, t_f^m) = I_{6x6}$

If $m+1 = MF$, $\Phi(t_A, t_f^m) = \Phi(t_o^{m+1}, t_o^m)$ with $m = MF$

If $m+1 < MF$, $\Phi(t_A, t_f^m) = \Phi(t_A, t_o^{m+1}) = \prod_{i=MF}^{m+1} \Phi(t_o^{i+1}, t_o^i)$

II.2.4 In-plane Matrix for Initial Point of Current Conic

With $E_k \triangleq E(t_o^m)$ or $F(t_o^m)$, employ equations in II.1.2b to obtain $\Phi(\tau_o^m, 0) \{ = \Phi(\tau_k, 0) \}$

II.2.5 $\Phi^{-1}(\tau_o^m, 0)$

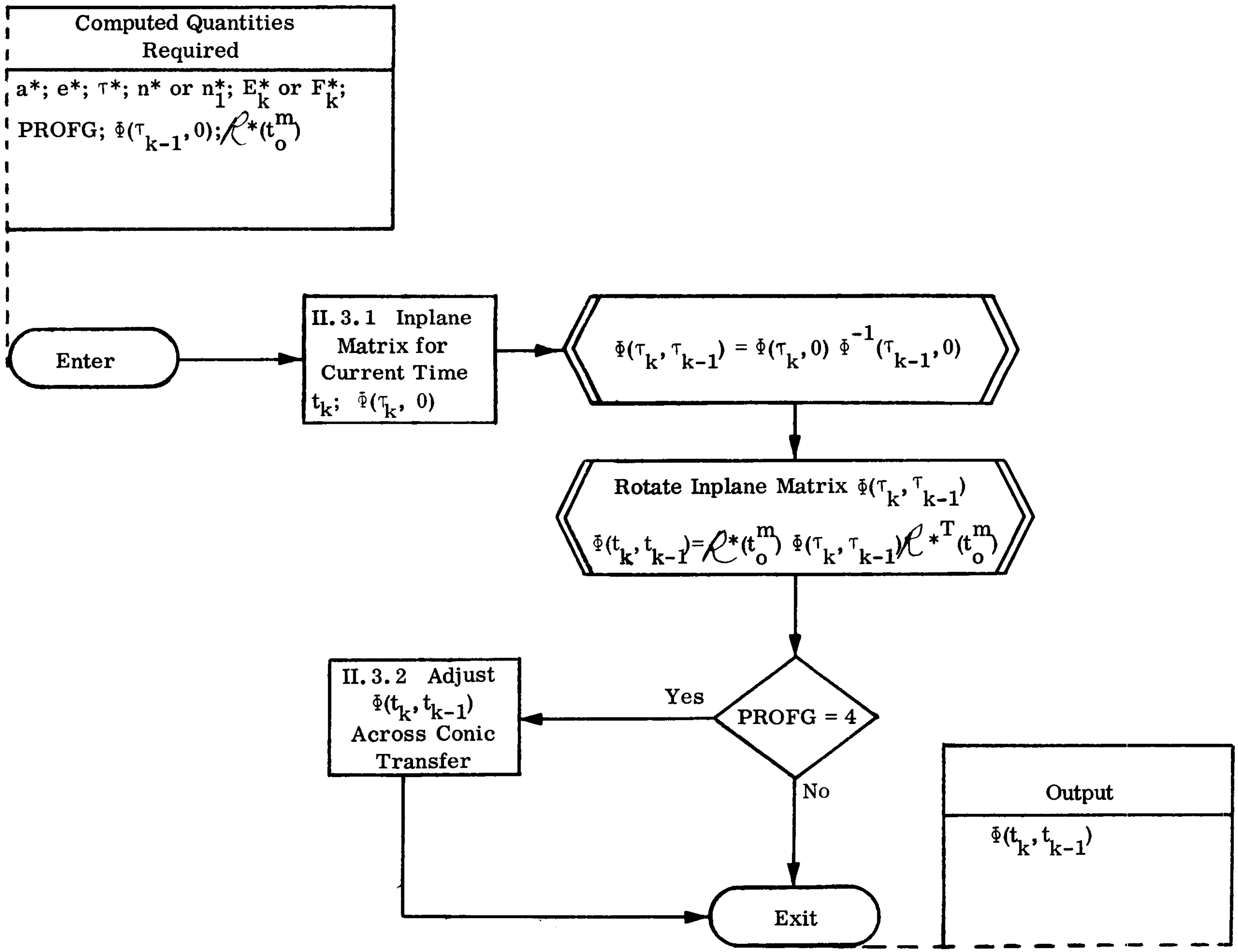
With $\Phi(\tau_o^m, 0) = \begin{bmatrix} \Phi_1 & \Phi_2 \\ \Phi_3 & \Phi_4 \end{bmatrix}$, $\Phi^{-1}(\tau_o^m, 0) = \begin{bmatrix} \Phi_4^T & -\Phi_2^T \\ -\Phi_3^T & \Phi_1^T \end{bmatrix}$

II.2.6 State Transition $\Phi(t_o^m, t_o)$

If $m = 1$, $\Phi(t_o^m, t_o) = I_{6x6}$

If $m = 2$, $\Phi(t_o^m, t_o) = \Phi(t_o^2, t_o^1)$

If $m > 2$, $\Phi(t_o^m, t_o) = \prod_{i=m}^{i=1} \Phi(t_o^i, t_o^{i-1})$



3.4.2.2.3 State Transition Between Observations - Block II.3

II.3.1 In-plane Matrix for Current Time t_k

Using E_k, employ equations in II.1.2b to obtain $\Phi(\tau_k, 0)$

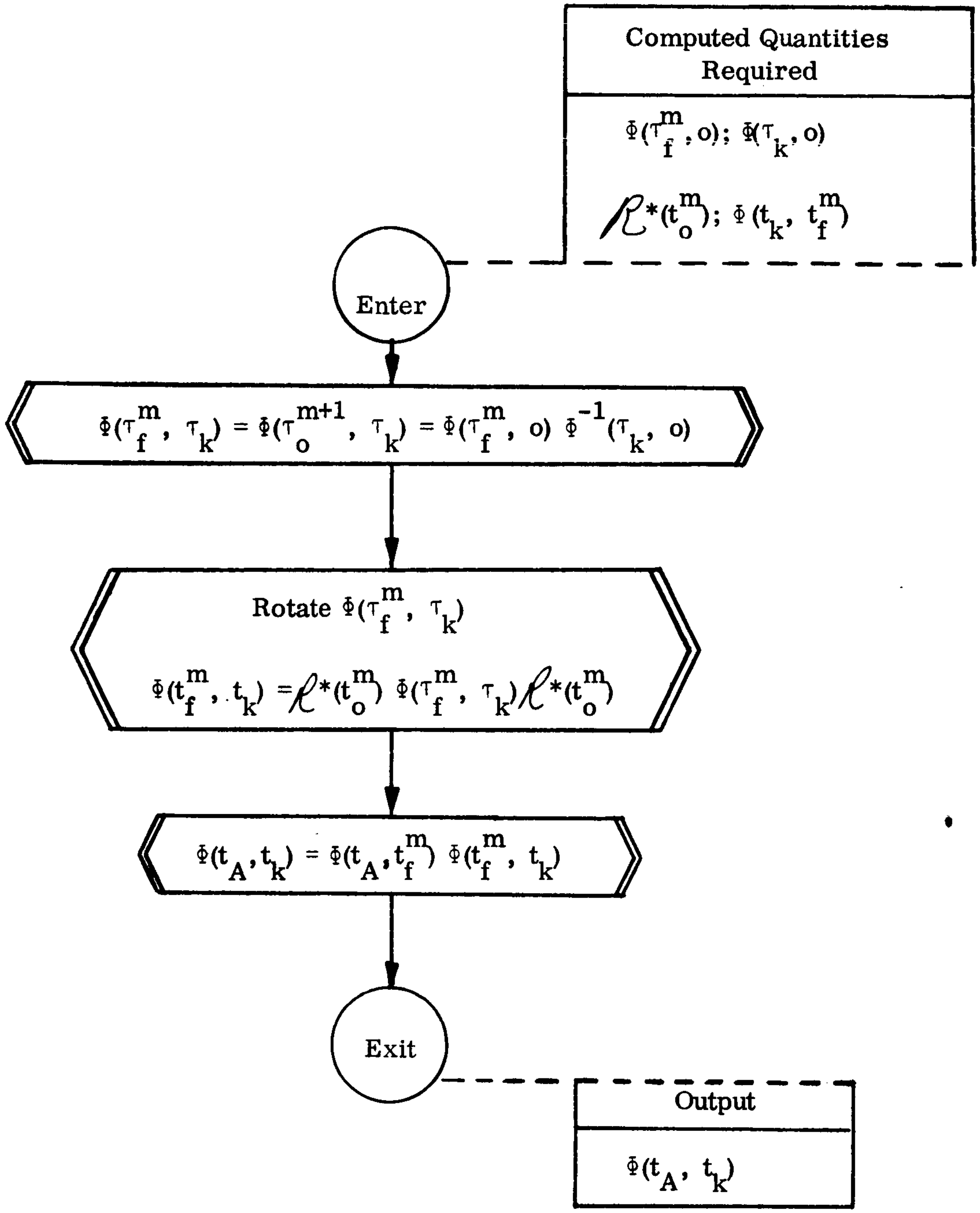
II.3.2 Adjust $\Phi(t_k, t_{k-1})$ Across Conic Breaks

If two consecutive observation times t_{k-1} and t_k fall on different conics, an artificial observation point is introduced at the conic transfer time t_o^m. Call this artificial time t_k'.

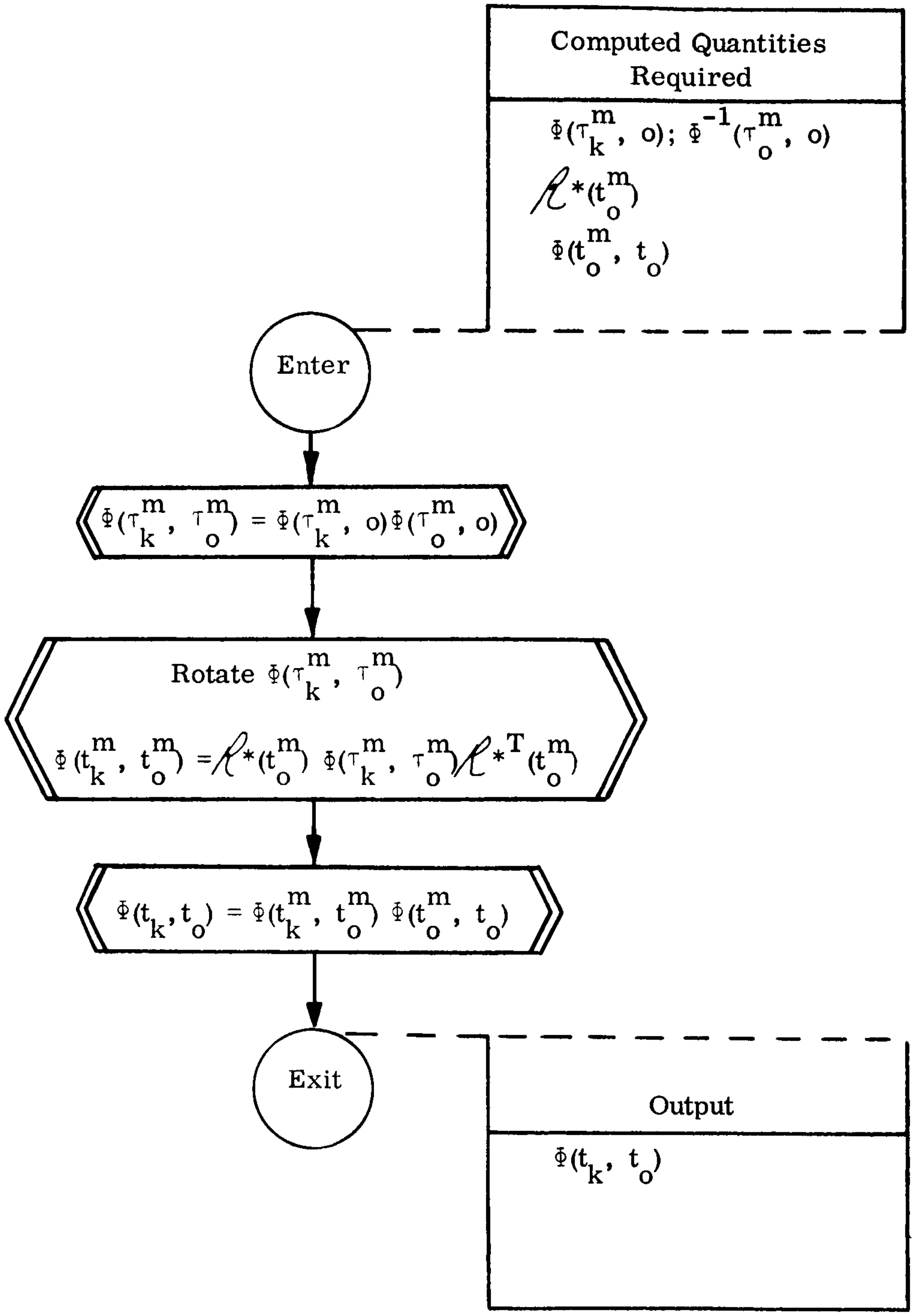
Now, what we have for $\Phi(t_k, t_{k-1})$ is really $\Phi(t_k, t_k')$ and what we need is

$$\Phi(t_k, t_{k-1}) = \Phi(t_k, t_k') \, \Phi(t_k', t_{k-1})$$

The second matrix is simply $\Phi(t_k', t_{k-1})$ computed at the artificial time $t_k = t_k'$ and was saved until now.

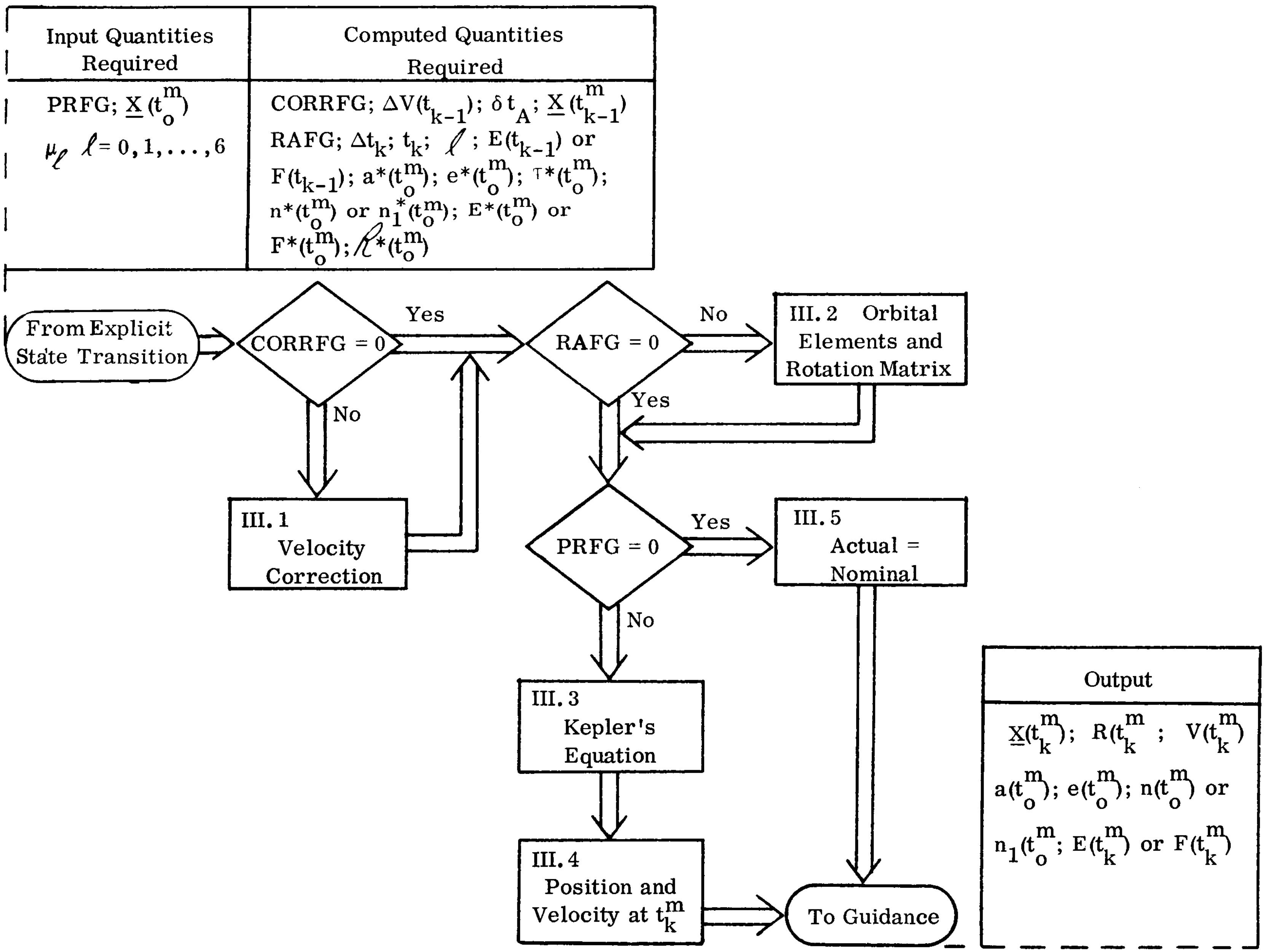


3.4.2.2.4 State Transition Between t_k and t_A - Block II.4



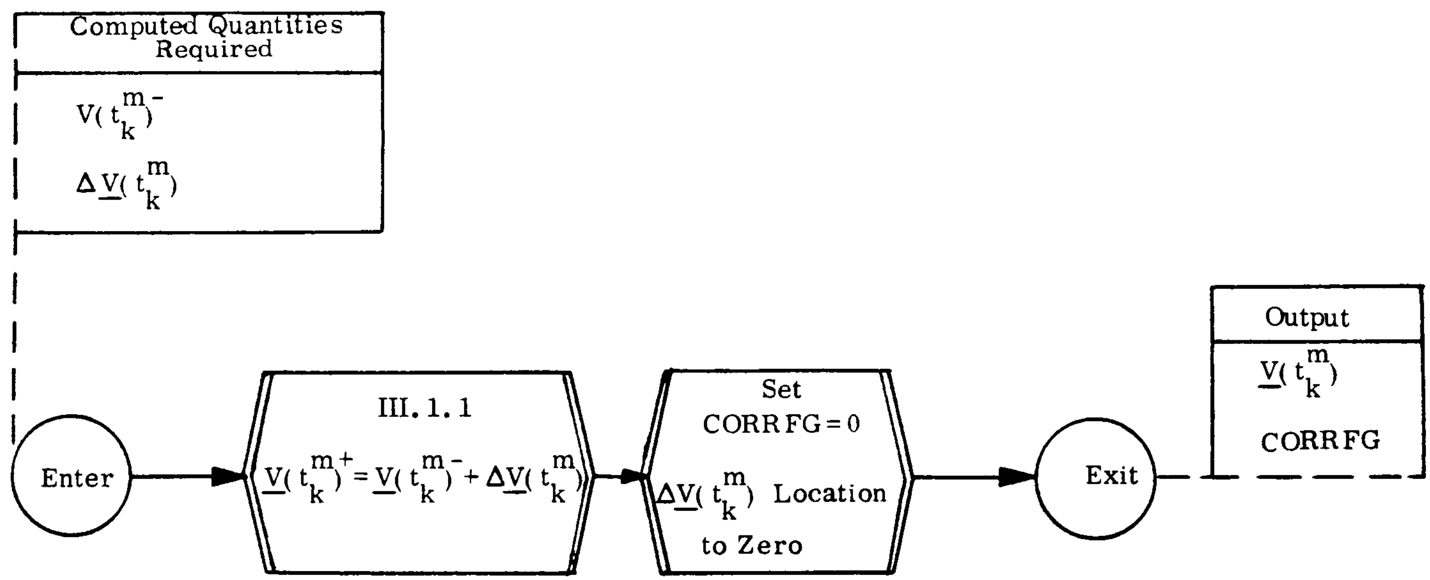
3.4.2.2.5 State Transition Between t_k and t_o - Block II.5

3.4.3 Two-Body Actual - Block III



3.4.3.1 Level II Flow Chart - Actual

3.4.3.2 Detailed Flow Charts and Equations



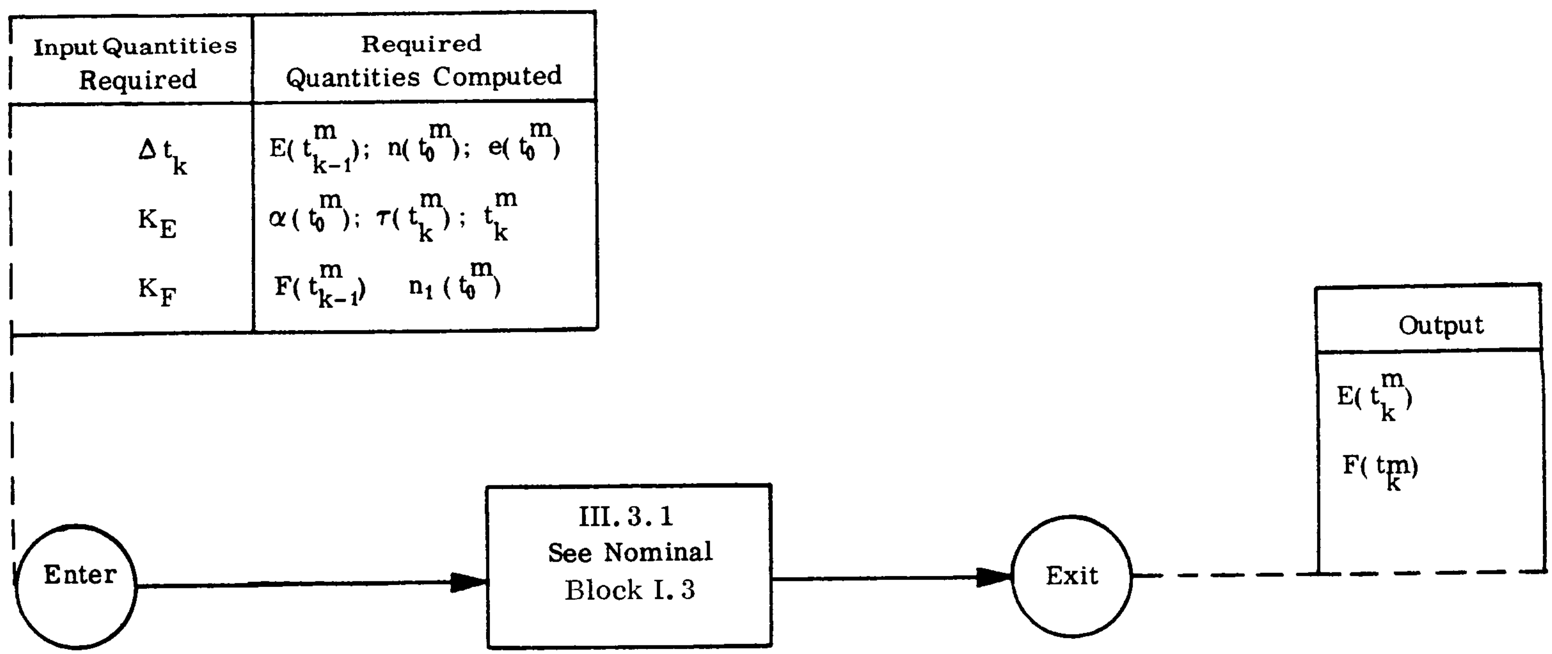
3.4.3.2.1 Velocity Correction - Block III.1

Input Quantities Required	Computed Quantities Required
μ_ℓ	$\alpha(t_k^m)$; $\underline{R}(t_k^m)$; $R(t_k^m)$ $V(t_k^m)$ $\underline{h}(t_k^m)$; $h(t_k^m)$ $\underline{V}(t_k^m)$

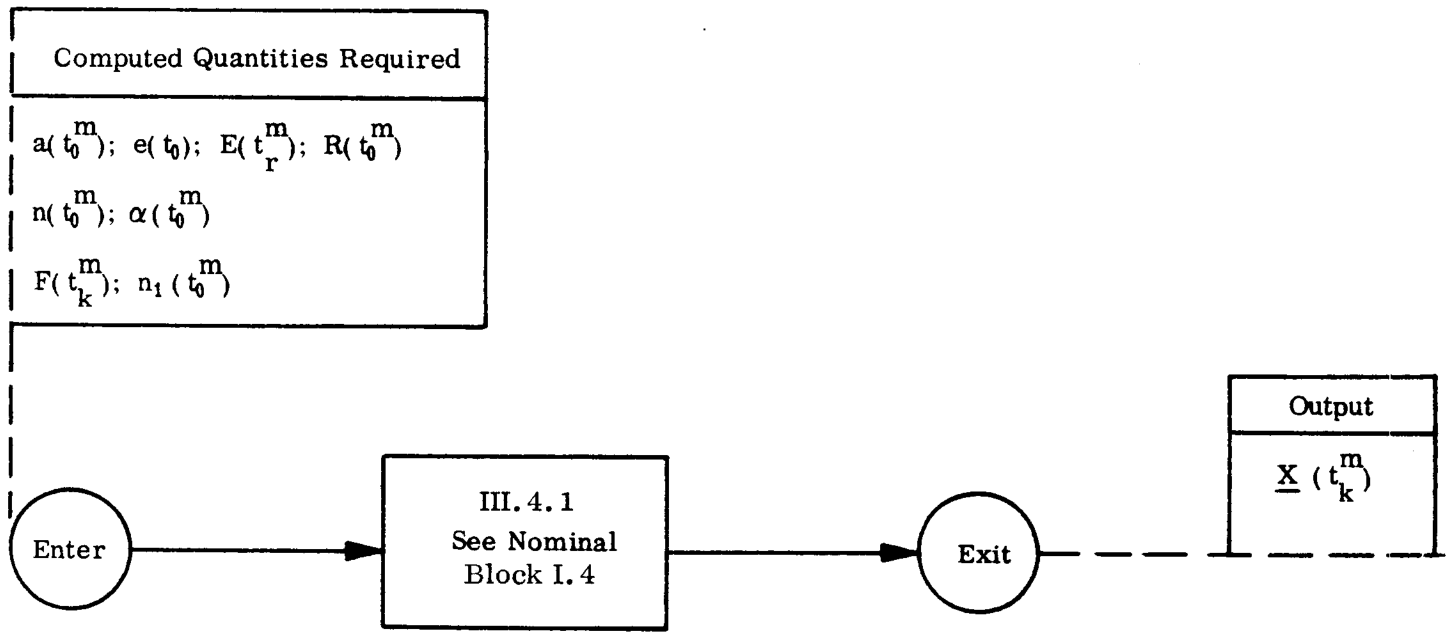
Enter → III. 2. 1 See Nominal Block I. 2 → Exit

Output
$\Omega(t_0^m)$: $E(t_0^m)$: $\tau(t_0^m)$ $e(t_0^m)$; $n(t_0^m)$; $a(t_0^m)$ $F(t_0^m)$; $n_1(t_0^m)$

3.4.3.2.2 Orbital Elements and Rotation Matrix - Actual - Block III. 2



3.4.3.2.3 Kepler's Equation - Block III. 3

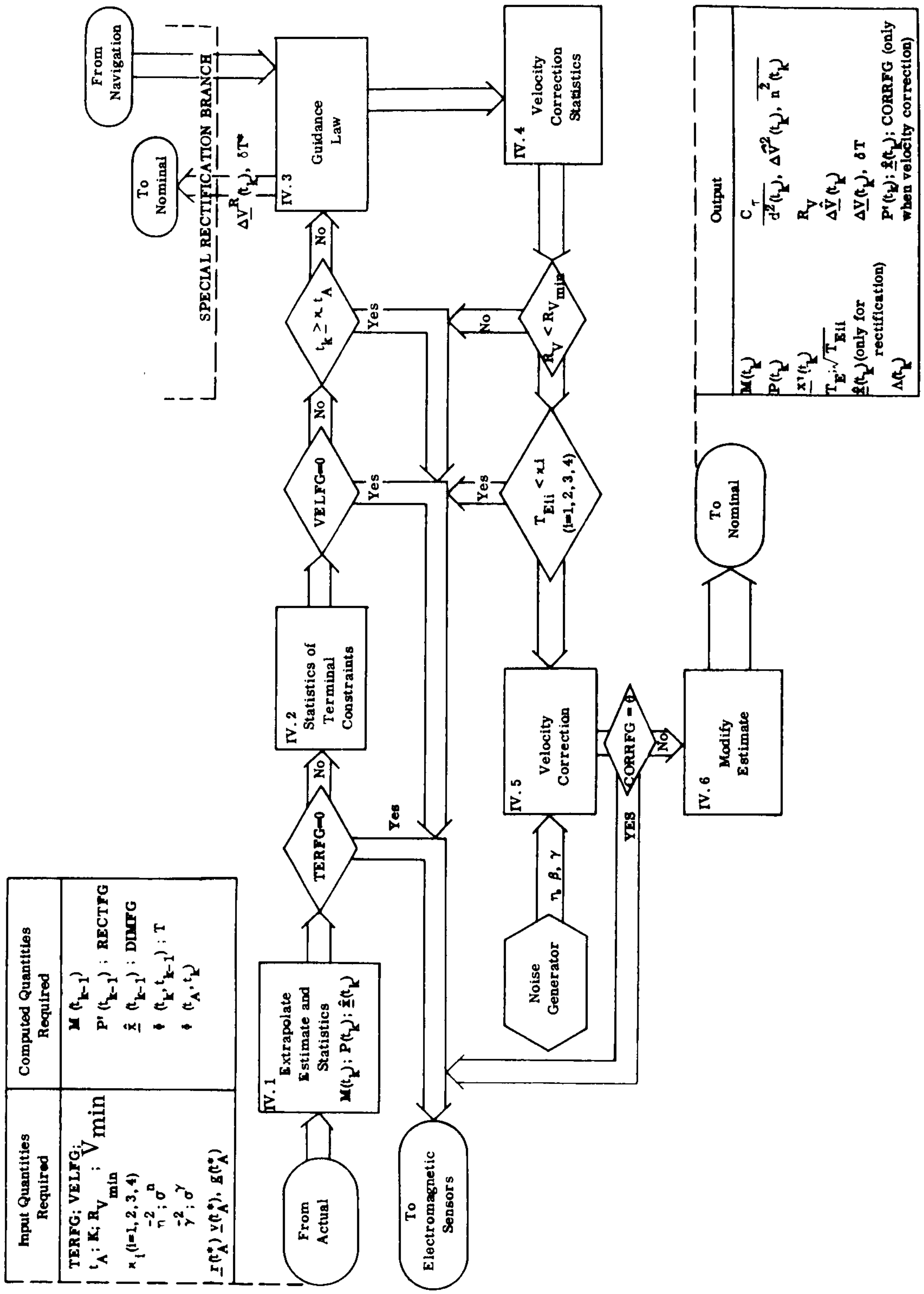


3.4.3.2.4 **Position and Velocity at t_k^m - Block III.4**

3.4.3.2.5 Actual = Nominal - Block III.5

In this block, the actual trajectory quantities required in subsequent blocks will be substituted by nominal quantities.

3.4.4 Guidance - Block IV

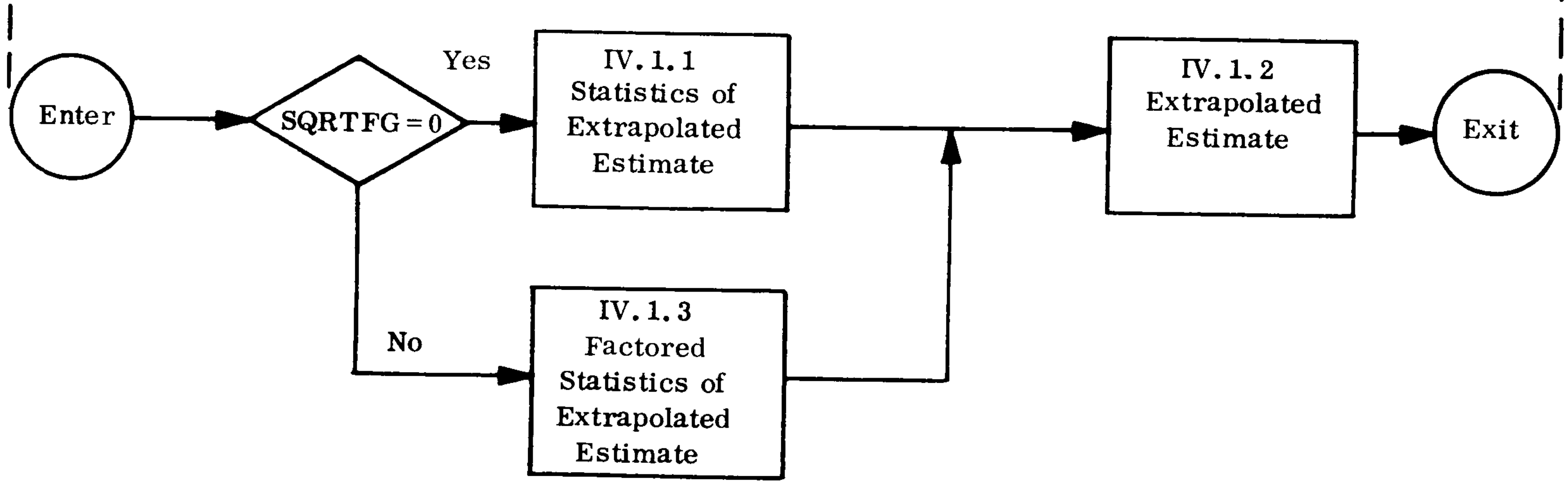


3.4.4.1 Level II Flow Chart - Guidance

3.4.4.2 Detailed Flow Charts and Equations

Input Quantities Required	Computed Quantities Required
SQRTFG	$M(t_{k-1})$ $P'_A(t_{k-1})$ or $\pi'_A(t_k)$ $\hat{\underline{x}}_A(t_{k-1})$ $\Phi(t_k, t_{k-1})$ DIMFG

Output
$M(t_k)$ $P_A(t_k)$ and/or $\pi_A(t_k)$ $\hat{\underline{x}}'_A(t_k)$



3.4.4.2.1 Extrapolation of Estimate and Statistics - Block IV. 1

IV. 1. 1 Statistics of Extrapolated Estimate

Compute the extrapolated statistics of the perturbation vector

$$M(t_k) = \Phi(t_k, t_{k-1})\, M(t_{k-1})\, \Phi^T(t_k, t_{k-1})$$

Now, introduce the following definitions of $\Phi_A(t_k, t_{k-1})$, $P_A(t_k)$, $P'_A(t_k)$ in terms of partitioned matrices.

$$\Phi_A(t_k, t_{k-1}) \underset{=}{\mathrm{Df}} \begin{bmatrix} \Phi(t_k, t_{k-1}) & O_1 \\ O_1^T & I \end{bmatrix}$$

where

$\Phi(t_k, t_{k-1})$	$\sim$	(6 x 6) state transition matrix computed in II.
O_1	$\sim$	6 x p matrix of zeros
I	$\sim$	p x p identify matrix

The number p is determined in Block B. It is defined by the second number of DIMFG.

$$P_A(t_k) \underset{=}{\mathrm{Df}} \begin{bmatrix} P(t_k) & P_1(t_k) \\ P_1^T(t_k) & P_2(t_k) \end{bmatrix}$$

$$P'_A(t_k) \underset{=}{\mathrm{Df}} \begin{bmatrix} P'(t_k) & P'_1(t_k) \\ P_1'^T(t_k) & P'_2(t_k) \end{bmatrix}$$

Then, in terms of these partitioned matrices, compute the extrapolated error in the estimate.

$$P_A(t_k) = \Phi_A(t_k, t_{k-1})\, P'_A(t_{k-1})\, \Phi_A(t_k, t_{k-1}) + Q_A(t_k)$$

where

$$Q_A(t_k) = \begin{bmatrix} Q(t_k) & 0 \\ 0 & 0 \end{bmatrix}$$ and $Q(t_k)$ is a (6x6) diagonal matrix

or

$$\begin{bmatrix} P(t_k)+Q(t_k) & P_1(t_k) \\ P_1^T(t_k) & P_2(t_k) \end{bmatrix} = \begin{bmatrix} \Phi(t_k, t_{k-1}) P'(t_{k-1}) \Phi^T(t_k, t_{k-1}) + Q(t_k) & \Phi(t_k, t_{k-1}) P_1'(t_{k-1}) \\ P_1'^T(t_{k-1}) \Phi^T(t_k, t_{k-1}) & P_2'(t_{k-1}) \end{bmatrix}$$

IV.1.2 Extrapolated Estimate

Define

$$\underline{\hat{x}}_A(t_k) \overset{Df}{=} \begin{bmatrix} \underline{\hat{x}}(t_k) \\ \underline{\hat{b}}(t_k) \end{bmatrix}$$

where $\underline{\hat{x}}(t_k)$ represents the first six components of the state vector

then,

$$\underline{\hat{x}}_A'(t_k) = \Phi_A(t_k, t_{k-1}) \underline{\hat{x}}_A(t_{k-1})$$

or,

$$\begin{bmatrix} \underline{\hat{x}}'(t_k) \\ \underline{\hat{b}}'(t_k) \end{bmatrix} = \begin{bmatrix} \Phi(t_k, t_{k-1}) \underline{\hat{x}}(t_{k-1}) \\ \underline{\hat{b}}(t_{k-1}) \end{bmatrix}$$

IV.1.3 Factored Statistics of the Error in the Extrapolated Estimate

Define the factored matrix

$$\pi_A'(t_k) = \begin{bmatrix} \pi'(t_k) & \pi_1'(t_k) \\ \pi_1'^T(t_k) & \pi_2'(t_k) \end{bmatrix}$$

then,

$$\pi_A(t_k) = \Phi_A(t_k, t_{k-1}) \pi_A'(t_{k-1})$$

or,

$$\begin{bmatrix} \pi(t_k) & \pi_1(t_k) \\ \pi_1^T(t_k) & \pi_2(t_k) \end{bmatrix} = \begin{bmatrix} \Phi(t_k, t_{k-1})\, \pi'(t_{k-1}) & \Phi(t_k, t_{k-1})\, \pi_1'(t_{k-1}) \\ \pi_1^T(t_{k-1}) & \pi_2'(t_k) \end{bmatrix}$$

Also,

$$P_A(t_k) = \pi_A(t_k)\, \pi_A^T(t_k)$$

and

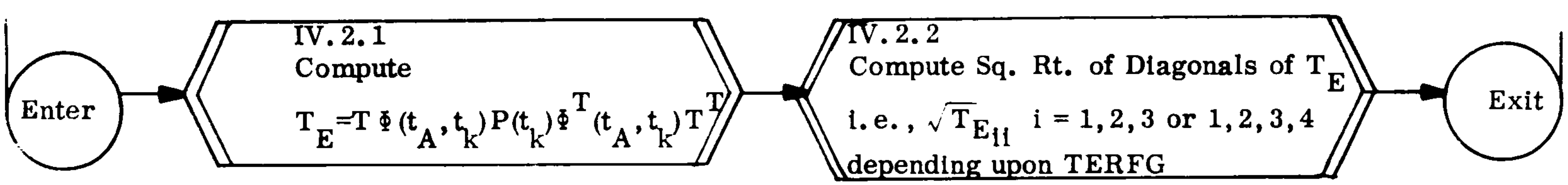
$$= \begin{bmatrix} \Phi\pi'\, \pi'^T \Phi^T + \Phi\pi_1'\, \pi_1'^T \Phi^T & \Phi\pi'\pi_1' + \Phi\pi_1'\, \pi_2'^T \\ (\Phi\pi'\, \pi_1' + \Phi\pi_1'\, \pi_2'^T)^T & \pi_1'^T\, \pi_1' + \pi_2'\, \pi_2'^T \end{bmatrix}$$

where the time indices have been omitted for notational simplicity.

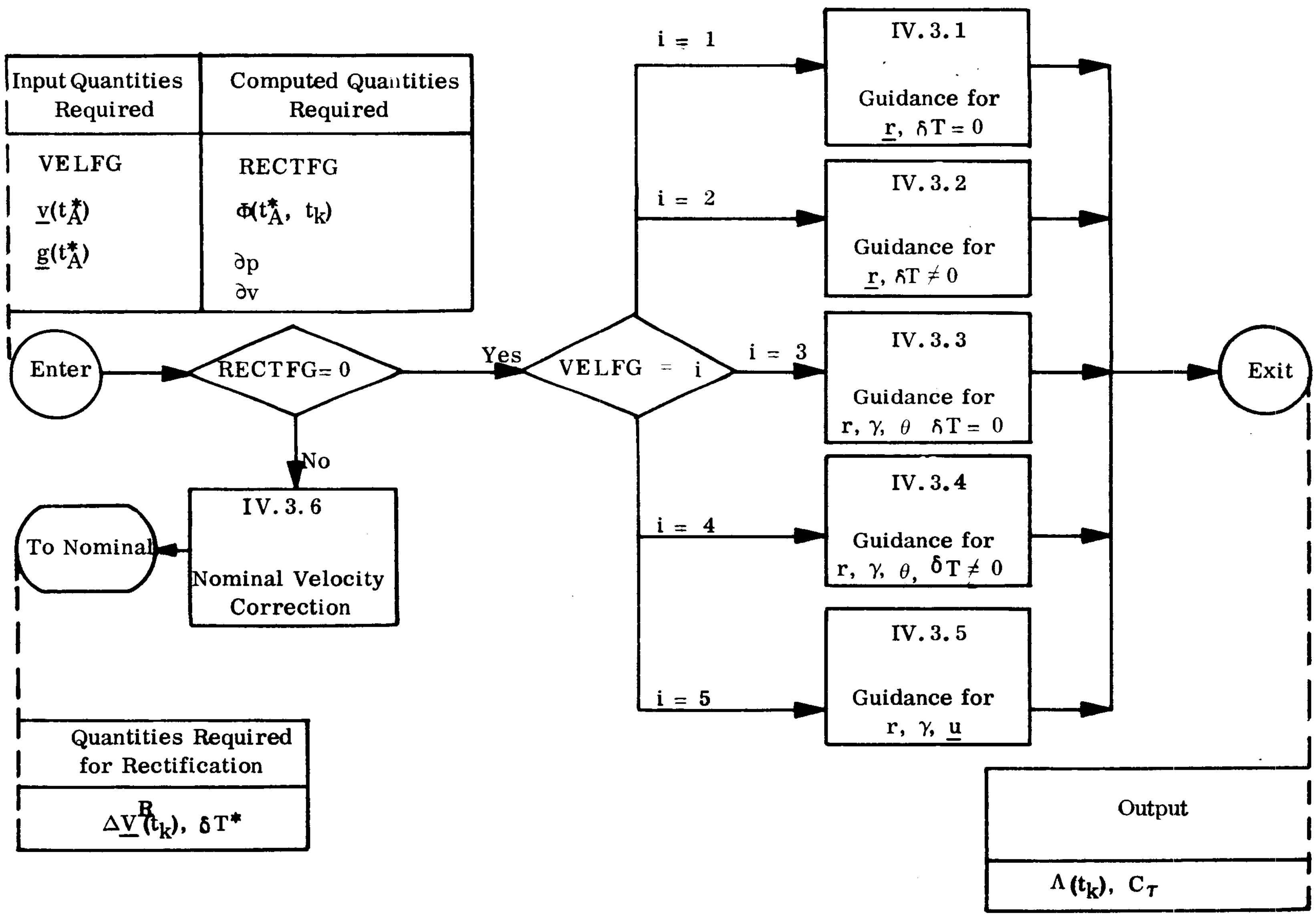
$$M(t_k) = \Phi(t_k, t_{k-1})\, M(t_{k-1})\, \Phi^T(t_k, t_{k-1})$$

Input Quantities Required	Computed Quantities Required
	T; $P(t_k)$ $\Phi(t_A, t_k)$

Output
T_E $\sqrt{T_{E_{ii}}}$



3.4.4.2.2 Statistics of Terminal Constraints - Block IV.2



3.4.4.2.3 Guidance Law - Block IV.3

IV.3.1 Guidance for $\underline{r}$, $\delta T = 0$

VELFG = 1

$$\Lambda(t_k) = -[\Phi_2^{-1}(t_A^*, t_k)\ \Phi_1(t_A^*, t_k) \quad I]$$

where

$\Phi(t_A^*, t_k)$ is the (6 x 6) transition matrix and

$$\Phi(t_A^*, t_k) = \begin{bmatrix} \Phi(t_A^*, t_k) & \Phi_1(t_A^*, t_k) \\ \Phi_2(t_A^*, t_k) & \Phi_3(t_A^*, t_k) \end{bmatrix}$$

$$C_\tau = 0$$

IV.3.2 Guidance for $\underline{r}$, $\delta T \neq 0$

VELFG = 2

$$\Lambda(t_k) = [I - \frac{\underline{\nu}_k \underline{\nu}_k^T}{\underline{\nu}_k^T \underline{\nu}_k}]\ \Lambda'(t_k)$$

where

$$\Lambda'(t_k) = -[\Phi_2^{-1}(t_A^*, t_k)\ \Phi_1(t_A^*, t_k) \quad I]$$

$$\underline{\nu}_k = -\Phi_2^{-1}(t_A^*, t_k)\ \underline{v}(t_A^*)$$

$$C_\tau = -\frac{\underline{\nu}_k^T \Lambda'(t_k)}{\underline{\nu}_k^T \underline{\nu}_k}$$

IV.3.3 Guidance for r, γ, θ, $\delta T = 0$

VELFG = 3

$$\Lambda(t_k) = -[A^{-1}B \quad I]$$

where

$$A = \partial p\ \Phi_2(t_A^*, t_k) + \partial v\ \Phi_4(t_A^*, t_k)$$

$$B = \partial p\ \Phi_1(t_A^*, t_k) + \partial v\ \Phi_3(t_A^*, t_k)$$

$$C_\tau \equiv 0$$

IV.3.4 Guidance for $r, \gamma, \theta,\ \delta T \neq 0$

VELFG = 4

$$\Lambda(t_k) = \left[I - \frac{\underline{\nu}_k \underline{\nu}_k^T}{\underline{\nu}_k^T \underline{\nu}_k}\right] \Lambda'(t_k)$$

where

$$\Lambda'(t_k) = -\left[A^{-1} B \quad I\right]$$

$$\underline{\nu}_k = -A^{-1} \underline{C}_c$$

$$\underline{C}_c \overset{\text{Df}}{\equiv} \begin{bmatrix} \dot{r}(t_A^*) \\ \dot{\gamma}(t_A^*) \\ \dot{\theta}(t_A^*) \end{bmatrix}$$

and

$$\dot{r}(t_A^*) = (\partial p^1)\ \underline{v}(t_A^*)$$

$$\dot{\gamma}(t_A^*) = (\partial p^2)\ \underline{v}(t_A^*) + (\partial v^2)\ \underline{g}(t_A^*)$$

$$\dot{\theta}(t_A^*) = (\partial v^3)\ \underline{g}(t_A^*)$$

$$C_\tau = \frac{-\underline{\nu}_k^T \Lambda'(t_k)}{\underline{\nu}_k^T \underline{\nu}_k}$$

IV.3.5 Guidance for $r, \gamma, \underline{u}$

VELFG = 5

$$\Lambda(t_k) = \Lambda'(t_k) - \underline{\nu}_k C_\tau$$

where $\Lambda'(t_k) = -[A^{-1}B \quad I]$

$\underline{\nu}_k = -A^{-1}\underline{C}_c$

where

$A = \partial p'\,\Phi_2(t^*_A, t_k) + \partial v'\,\Phi_4(t^*_A, t_k)$; $B = \partial p'\,\Phi_1(t^*_A, t_k) + \partial v'\,\Phi_3(t^*_A, t_k)$

$$\partial p' = \begin{bmatrix} \partial p^1 \\ \partial p^2 \\ \partial p^3 \end{bmatrix} ; \qquad \partial v' = \begin{bmatrix} \partial v^1 \\ \partial v^2 \\ \partial v^3 \end{bmatrix}$$

$$\underline{C}_c = \begin{bmatrix} \dot{r}(t^*_A) \\ \dot{\gamma}(t_A) \\ \dot{u}_x(t^*_A) \end{bmatrix}$$

and $\dot{r}(t^*_A) = \partial p^1\,\underline{v}(t^*_A)$

$\dot{\gamma}(t^*_A) = \partial p^2\,\underline{v}(t^*_A) + \partial v^2\,\underline{g}(t^*_A)$

$$\dot{u}_x(t^*_A) = \frac{1}{|\underline{r}(t^*_A) \times \underline{v}(t^*_A)|}\{\underline{r}(t^*_A) \times \underline{g}(t^*_A) - \underline{u}(t^*_A)[\underline{u}(t^*_A)\cdot \underline{r}(t^*_A) \times \underline{g}(t^*_A)]\}_{x\text{-comp.}}$$

$$C_\tau = \frac{1}{\dot{u}_y(t^*_A) + \underline{a}^T\,\underline{\nu}_k}[\underline{b}^T - \underline{a}^T A^{-1} B][I \quad 0]$$

where $\dot{u}_y(t^*_A) = \dfrac{1}{|\underline{r}(t^*_A) \times \underline{v}(t^*_A)|}\{\underline{r}(t^*_A) \times \underline{g}(t^*_A) - \underline{u}(t^*_A)[\underline{u}(t^*_A)\cdot \underline{r}(t^*_A) \times \underline{g}(t^*_A)]\}_{y\text{-comp}}$

$\underline{a}^T = \partial p^4\,\Phi_2(t^*_A, t_k) + \partial v^4\,\Phi_4(t^*_A, t_k)$

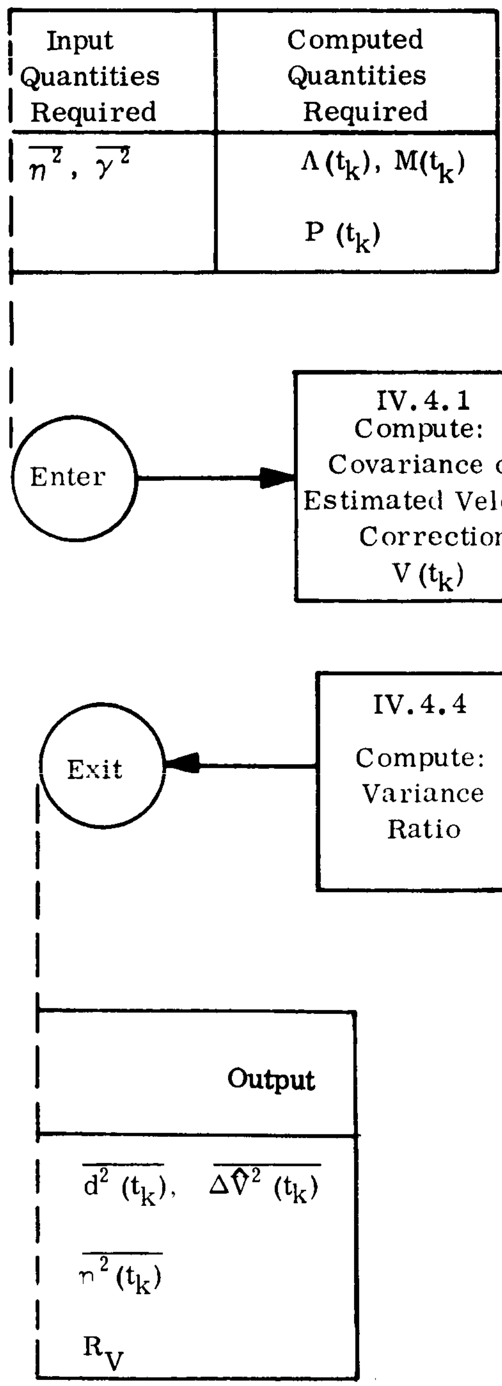
$\underline{b}^T = \partial p^4\,\Phi_1(t^*_A, t_k) + \partial v^4\,\Phi_3(t^*_A, t_k)$

IV.3.6 Compute Nominal Velocity Correction

$$\Delta\underline{V}^R(t_k) = \Lambda(t_k)\,\hat{\underline{x}}(t_k)$$

$$\delta T^* = C_\tau\,\hat{\underline{x}}(t_k)$$

(This branch should only be traversed for orbit rectification. In this case the entry would have been from the navigation block so the current estimate should be available.)



3.4.4.2.4 Velocity Correction Statistics - Block IV.4

IV.4.1 Covariance of Estimated Velocity Correction

$$V(t_k) = \Lambda(t_k)[M(t_k) - P(t_k)]\Lambda^T(t_k)$$

where $M(t_k)$ and $P(t_k)$ and the (6x6) submatrices of upper left-hand corner of $M_A(t_k)$ and $P_A(t_k)$.

IV.4.2 Covariance of Error in Estimated Correction

$$D(t_k) = \Lambda(t_k) P(t_k) \Lambda^T(t_k)$$

IV.4.3 Covariance of Error in Actual Correction

$$N(t_k) = \overline{\eta^2}\, V(t_k) + \frac{\overline{\gamma^2}}{2}\, [v(t_k) I - V(t_k)]$$

where $v(t_k) = \sum_i \sum_j \beta_{ij}(t_k)[m_{ij}(t_k) - p_{ij}(t_k)]$

and

$$M(t_k) = \{m_{ij}(t_k)\}$$

$$P(t_k) = \{p_{ij}(t_k)\}$$

$$\Lambda^T(t_k)\Lambda(t_k) = \{\beta_{ij}(t_k)\}$$

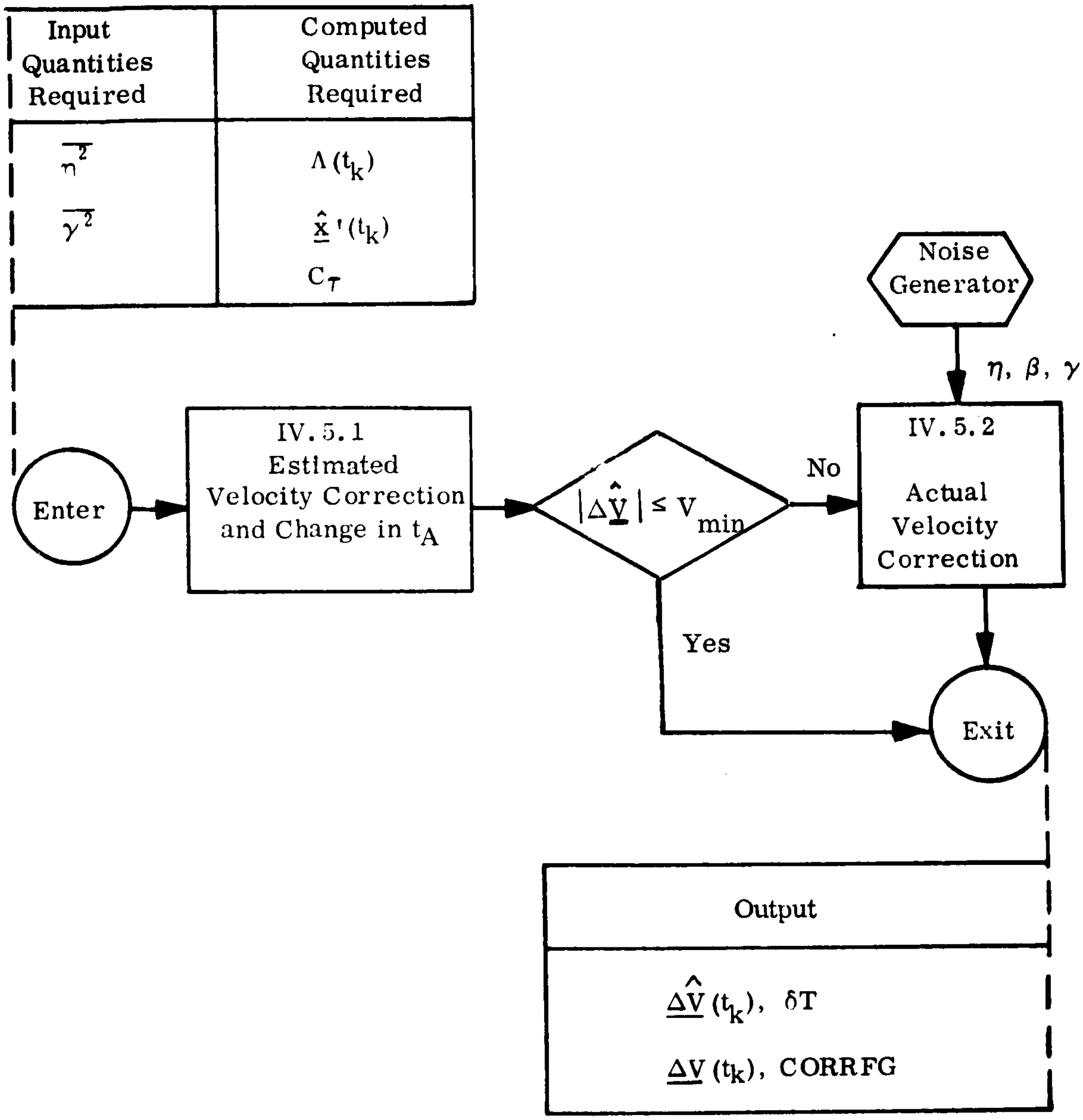
IV.4.4 Variance Ratio

$$\overline{d^2(t_k)} = \text{trace } D(t_k)$$

$$\overline{\Delta\hat{V}^2(t_k)} = \text{trace } V(t_k)$$

$$\overline{n^2(t_k)} = \text{trace } N(t_k)$$

$$R_V = \frac{\overline{d^2(t_k)}}{\overline{\Delta\hat{V}^2(t_k)}}$$



3.4.4.2.5 Velocity Correction - Block IV.5

IV.5.1 Estimated Velocity Correction and Change in Time of Arrival

$$\Delta \hat{\underline{V}}(t_k) = \Lambda(t_k)\, \hat{\underline{x}}'(t_k)$$

$$\delta T = C_T\, \hat{\underline{x}}'(t_k)$$

where $\hat{\underline{x}}'(t_k)$ is (6 x 1).

IV.5.2 Actual Velocity Correction

Set CORRFG $\neq 0$

$$M_T = [\underline{1}'_x\ \underline{1}'_y\ \underline{1}'_z]$$

where

$$\underline{1}'_x = \frac{\Delta \hat{\underline{V}}(t_k)}{\Delta \hat{V}(t_k)}$$

$$\underline{1}'_y = \underline{1}'_z \times \underline{1}'_x$$

$$\underline{1}'_z = \frac{\Delta \underline{N}(t_k)}{\Delta N(t_k)}$$

and

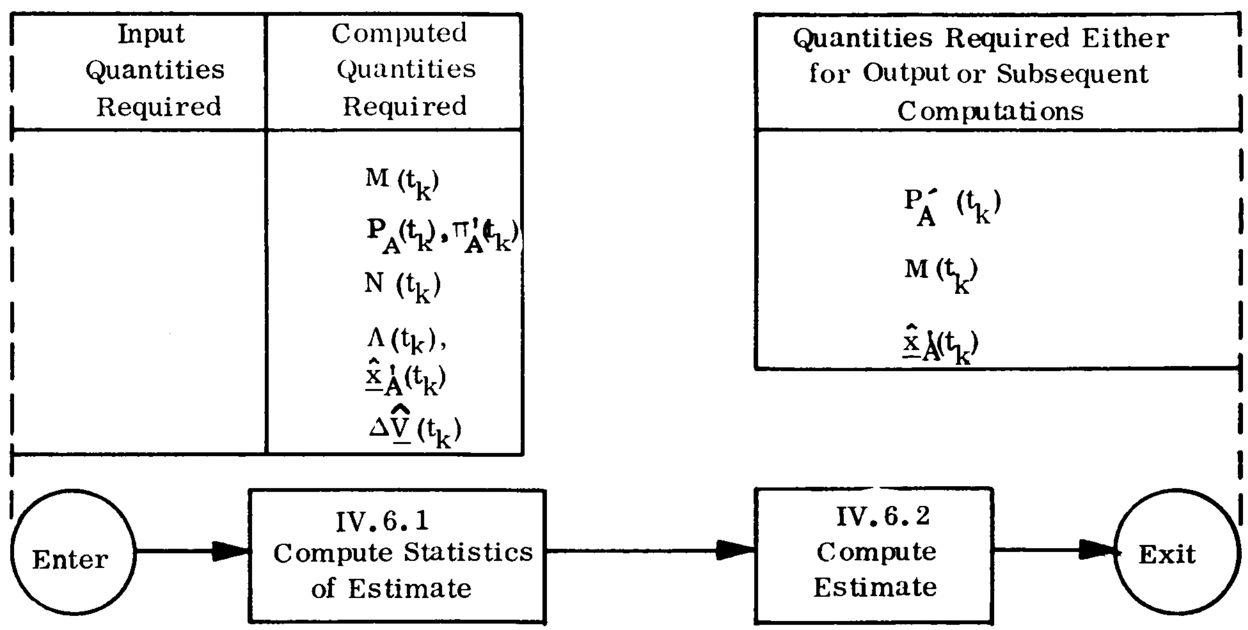
$$\Delta \underline{N}(t_k) = \begin{bmatrix} 0 \\ -\Delta \hat{V}_z \\ \Delta \hat{V}_y \end{bmatrix}$$

$$\Delta \underline{V}(t_k) = (1 + \eta)\, \Delta \hat{V}(t_k)\, M_T \begin{bmatrix} 1 \\ \gamma \cos \beta \\ \gamma \sin \beta \end{bmatrix}$$

The γ and η are obtained from a gaussian noise generator and have mean zero and variances

$$(1 + \sigma^n)\, \overline{\eta^2}; \quad (1 + \sigma^\gamma)\, \overline{\gamma^2}$$

The angle β is random number that is uniformly distributed for $\beta \in [-\pi, \pi]$.



3.4.4.2.6 Modify Estimate - Block IV.6

IV.6.1 Statistics of Estimate

$$P'_A(t_k) = P_A(t_k) + JN(t_k)J^T$$

or

$$P'_A(t_k) = \Pi_A(t_k)\Pi_A^T(t_k) + JN(t_k)J^T$$

and when SQRTFG $\neq 0$, $P'_A(t_k)$ has to be refactored via triangularization routine to obtain $\Pi'_A(t_k)$.

$$J = \begin{bmatrix} J_1 \\ \\ 0_1 \end{bmatrix} = \begin{bmatrix} 0 \\ I \\ 0_1 \end{bmatrix} ; \quad \begin{array}{lll} 0 & \sim & \text{(3 x 3) zero matrix} \\ I & \sim & \text{(3 x 3) identity matrix} \\ 0_1 & \sim & \text{p x 3 matrix of zeros} \end{array}$$

$$M'(t_k) = [I + J_1\Lambda(t_k)][M(t_k) - P(t_k)][I + J_1\Lambda(t_k)]^T + P(t_k) + J_1N(t_k)J_1^T$$

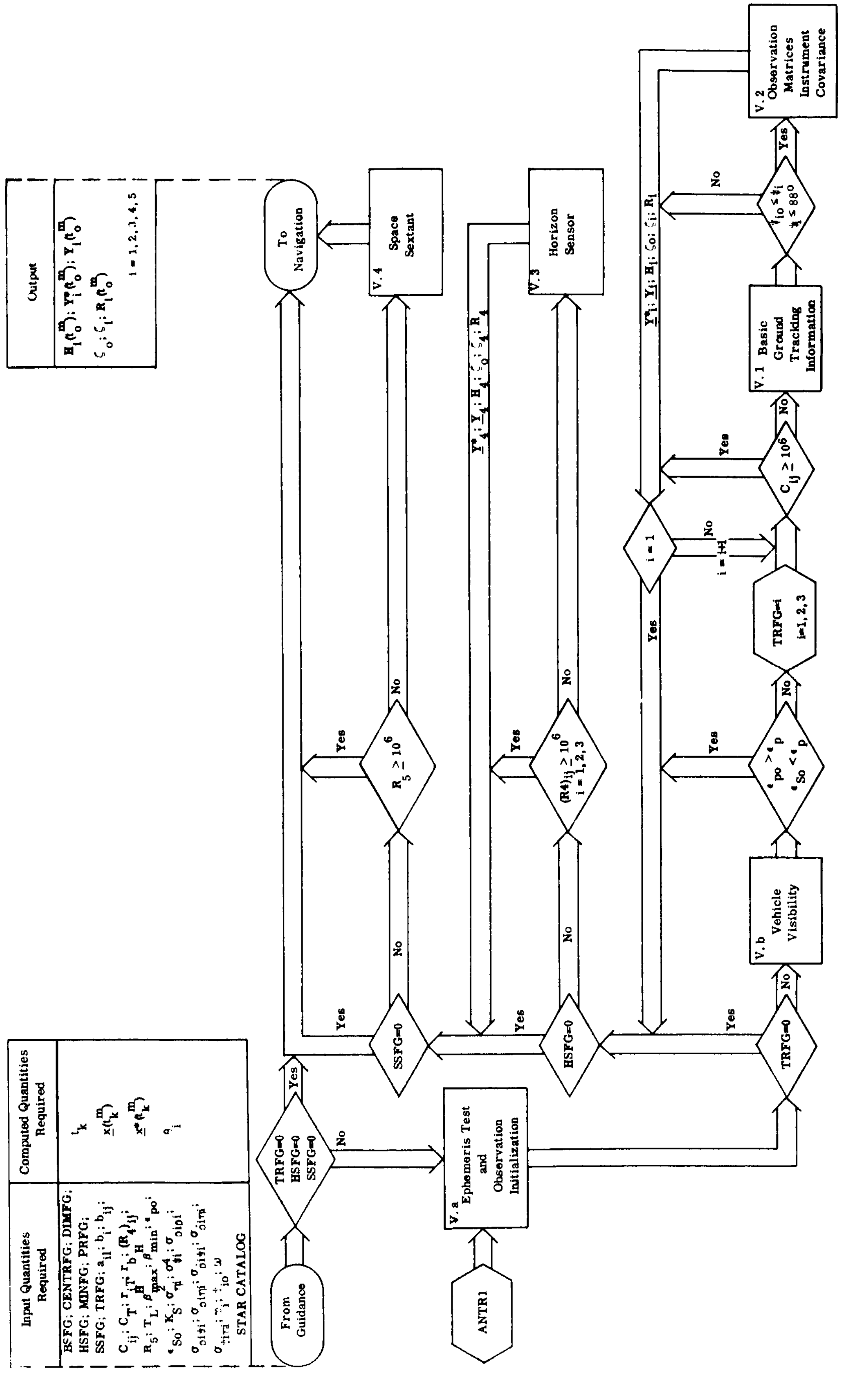
Then, set

$$M(t_k) = M'(t_k)$$

IV.6.2 Estimate

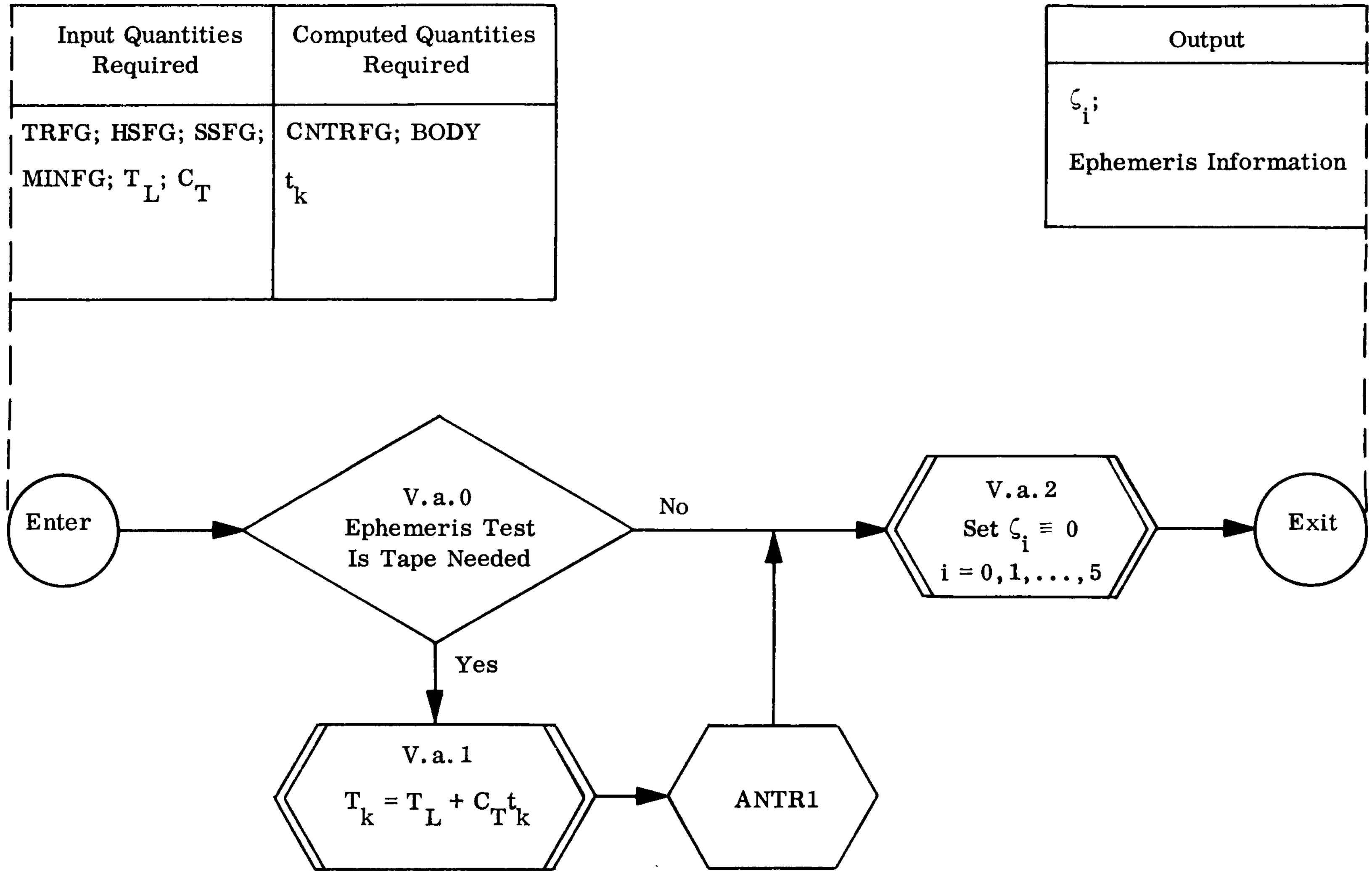
$$\hat{\underline{x}}_A(t_k) = \hat{\underline{x}}'_A(t_k) + J\,\Delta\hat{\underline{V}}(t_k)$$

3.4.5 Electromagnetic Sensors - Block V



3.4.5.1 Level II Flow Chart - Electromagnetic Sensors

3.4.5.2 Detailed Flow Charts and Equations



3.4.5.2.1 Ephemeris Tape and Observation Initialization - Block V.a

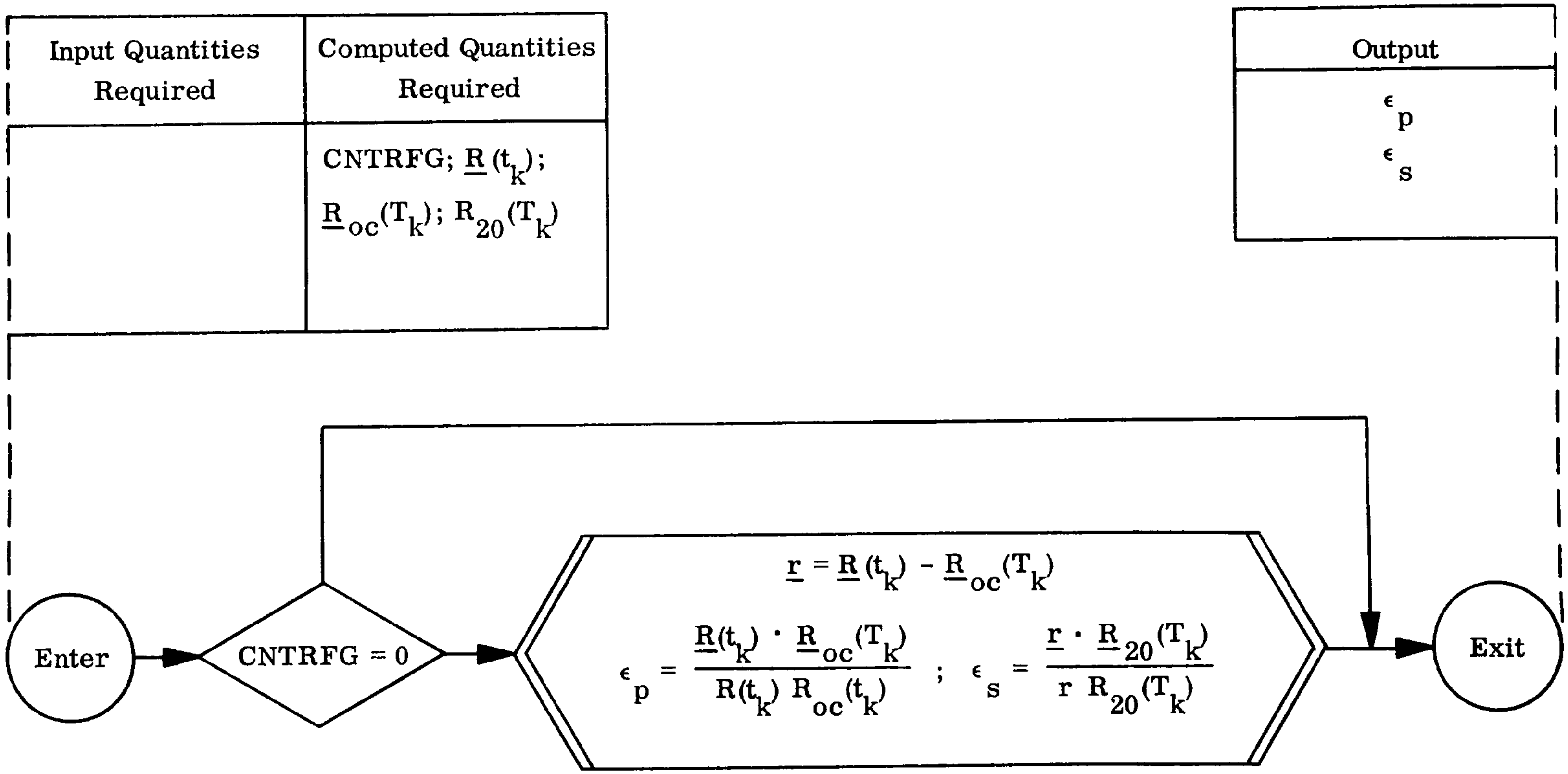
V.a.0 Ephemeris Test

1. If TRFG $\neq$ 0 and CNTRFG $\neq$ 0 = Need Ephemeris Tape

2. If HSFG $\neq$ 0 and CNTRFG $\neq$ BODY = Need Ephemeris Tape

3. If SSFG $\neq$ 0 and MINFG = 2 = Need Ephemeris Tape

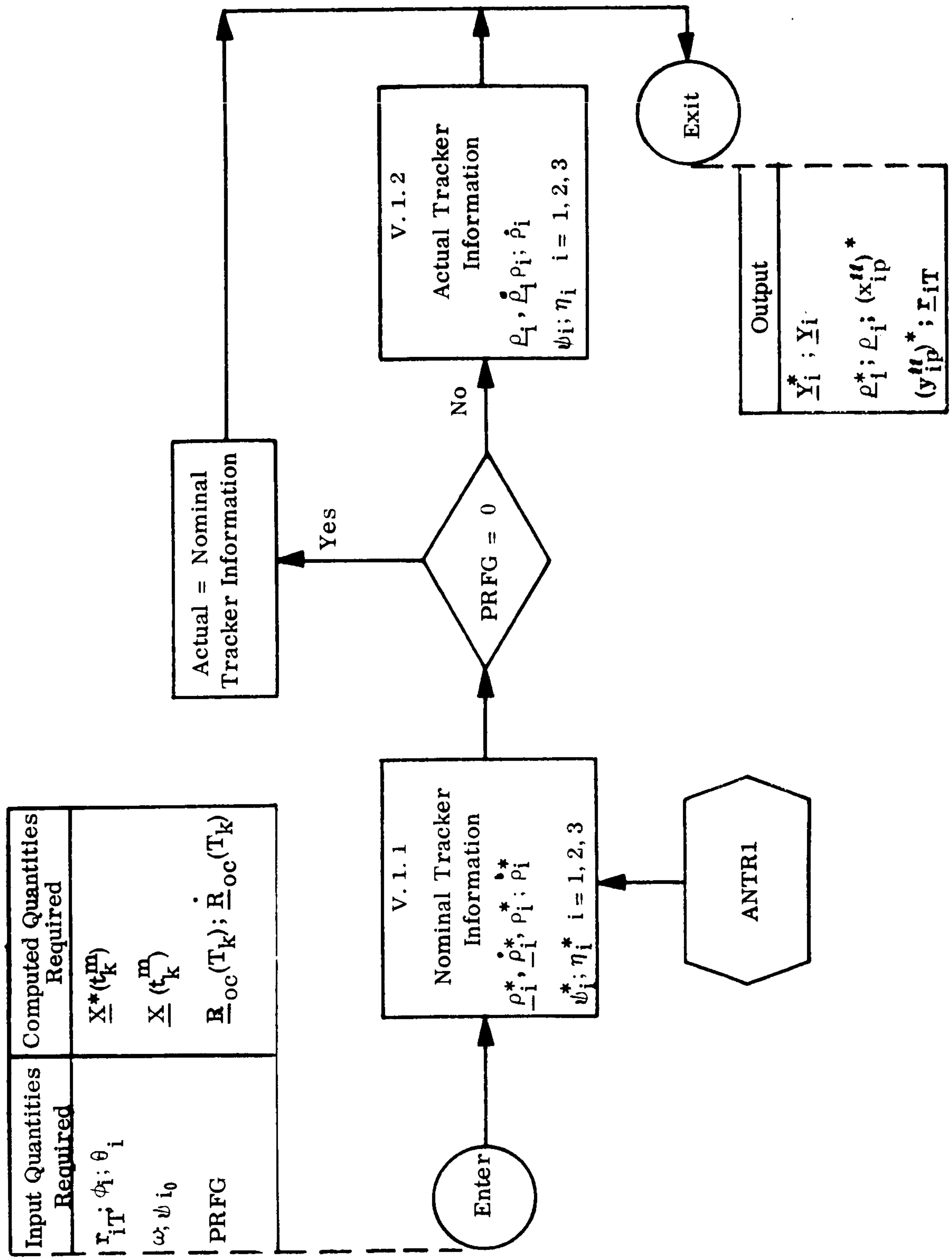
4. If SSFG $\neq$ 0, MINFG $= \left\{ \begin{matrix} 0 \\ 1 \end{matrix} \right.$ and CNTRFG $\neq$ BODY = Need Ephemeris Tape.



$\underline{R}_{oc}$ = Position of Earth with respect to central body c.

$\underline{R}_{20}$ = Position of Sun with respect to Earth.

3.4.5.2.2 Vehicle Visibility Criteria - Block V.b



3.4.5.2.3 Basic Ground Tracking Information - Block V.1

V.1.1 Nominal Tracker Information

The range and range rate vector equations are

$$\underline{\rho}_i^*(t_k) = \underline{R}^*(t_k) - \underline{r}_{iT}(t_k) - \underline{R}_{oc}(T_k)$$

$$\underline{\dot{\rho}}_i^*(t_k) = \underline{\dot{R}}^*(t_k) - \underline{\dot{r}}_{iT}(t_k) - \underline{\dot{R}}_{oc}(T_k)$$

where $\underline{R}_{oc}$ and $\underline{\dot{R}}_{oc}$ are the position and velocity of the earth with respect to the central body (in equatorial non-rotating coordinates) obtained from ephemeris subroutine ANTR1. $\underline{R}^*$ and $\underline{\dot{R}}^*$ are the vehicle position and velocity w.r.t. the central body, i.e.,

$$\underline{R}^*(t_k) = \begin{bmatrix} X_1^*(t_k) \\ X_2^*(t_k) \\ X_3^*(t_k) \end{bmatrix} \quad ; \quad \underline{\dot{R}}(t_k) = \begin{bmatrix} X_4^*(t_k) \\ X_5^*(t_k) \\ X_6^*(t_k) \end{bmatrix}$$

and

$$\underline{r}_{iT} = \begin{bmatrix} X_{iT}(t_k) \\ Y_{iT}(t_k) \\ Z_{iT}(t_k) \end{bmatrix} = \begin{bmatrix} r_{iT} \cos \varphi_i \cos (\theta_i + \omega t_k) \\ r_{iT} \cos \varphi_i \sin (\theta_i + \omega t_k) \\ r_{iT} \sin \varphi_i \end{bmatrix} \qquad i = 1, 2, 3$$

$$\underline{\dot{r}}_{iT} = \begin{bmatrix} -\omega Y_{iT}(t_k) \\ \omega X_{iT}(t_k) \\ 0 \end{bmatrix}$$

Define inertial probe (w.r.t. tracker) positions

$$\underline{\rho}_i^* = \begin{bmatrix} X_{ip}^* \\ Y_{ip}^* \\ Z_{ip}^* \end{bmatrix}$$

Then the range and range rate from the trackers are

$$\rho_i^* = [X_{ip}^2 + Y_{ip}^2 + Z_{ip}^2]^{1/2}$$

$$\dot{\rho}_i^* = \frac{\underline{\rho}_i^{*T} \dot{\underline{\rho}}_i^*}{\rho_i^*}$$

Elevation and azimuth angle of probe w.r.t. tracker

$$\psi_i^* = \sin^{-1}\left[\frac{\underline{r}_{iT}^T \underline{\rho}_i^*}{r_{iT}\,\rho_i^*}\right] \qquad -90^\circ < \psi_i < 90^\circ$$

$$\eta_i^* = \begin{cases} \cos^{-1}\left[\dfrac{-x''^*_{ip}}{\rho_i^* \cos \psi_i^*}\right] \\ \sin^{-1}\left[\dfrac{y''^*_{ip}}{\rho_i^* \cos \psi_i^*}\right] \end{cases} \qquad 0 \le \eta_i \le 360^\circ$$

where

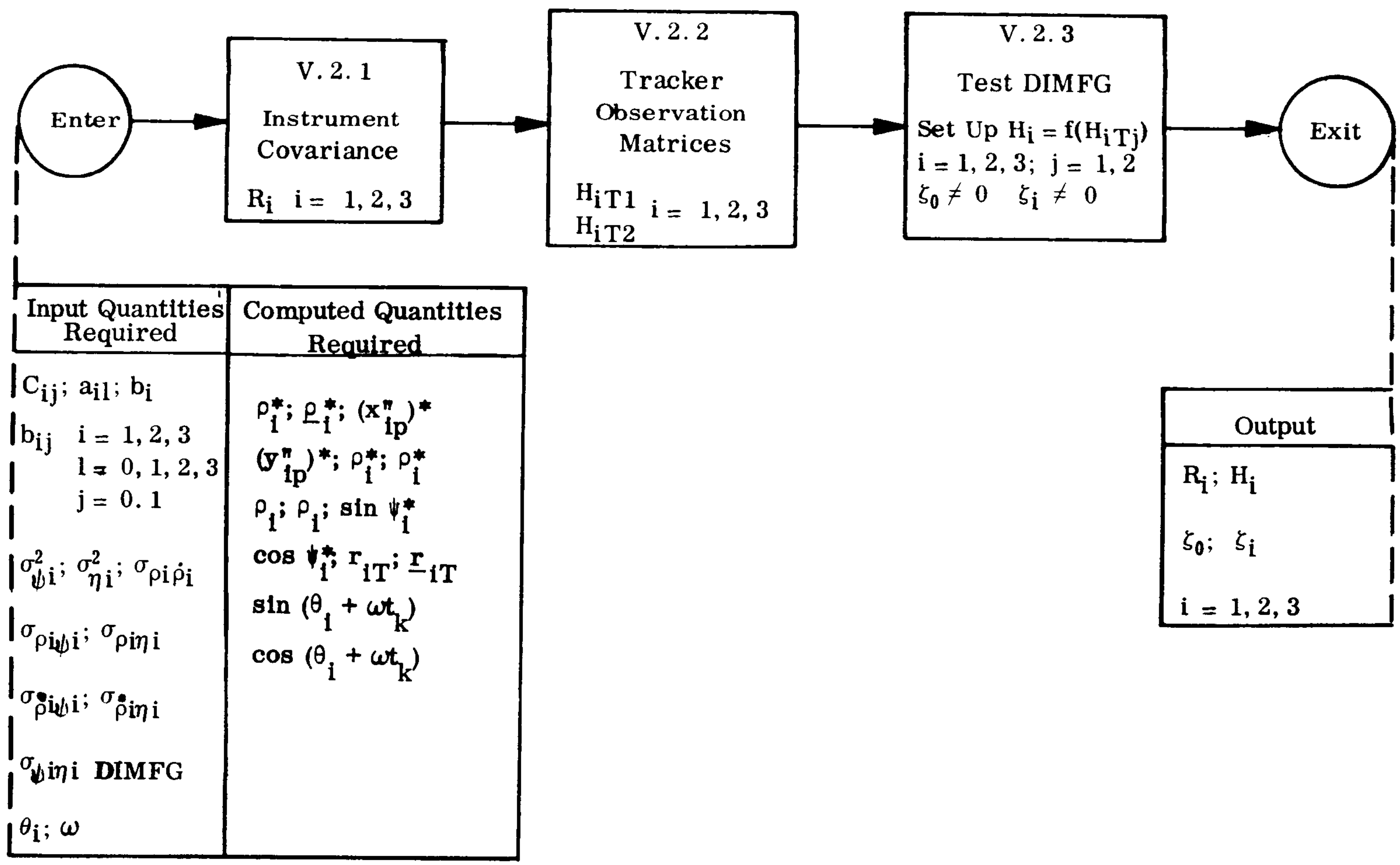
$$\begin{bmatrix} x''^*_{ip} \\ y''^*_{ip} \end{bmatrix} = \begin{bmatrix} \sin\varphi_i \cos(\theta_i + \omega t_k) \vdots \sin\varphi_i \sin(\theta_i + \omega t_k) \vdots - \cos\varphi_i \\ -\sin(\theta_i + \omega t_k) \qquad \cos(\theta_i + \omega t_k) \qquad 0 \end{bmatrix} \begin{bmatrix} X^*_{ip} \\ Y^*_{ip} \\ Z^*_{ip} \end{bmatrix}$$

$$\underline{Y}_i^* \overset{Df}{=} \begin{bmatrix} \rho_i^* \\ \dot{\rho}_i^* \\ \psi_i^* \\ \eta_i^* \end{bmatrix} \qquad i = 1,2,3$$

V.1.2 Actual Tracker Information

The equations for the actual values are the same as in V.1.1 except that $\underline{X}(t_k) \rightarrow \underline{X}^*(t_k)$

$$\underline{Y}_i \overset{Df}{=} \begin{bmatrix} \rho_i \\ \rho_i \\ \psi_i \\ \eta_i \end{bmatrix}$$



3.4.5.2.4 Observation Matrices and Instrument Covariance for Ground Tracking - Block V.2

V.2.1 Instrument Covariance

$$R_i = C_{ij} \begin{bmatrix} \sigma^2_{\rho i} & \sigma_{\rho i \dot{\rho} i} & \sigma_{\rho i \psi i} & \sigma_{\rho i \eta i} \\ & \sigma^2_{\dot{\rho} i} & \sigma_{\dot{\rho} i \psi i} & \sigma_{\dot{\rho} i \eta i} \\ & & \sigma^2_{\psi i} & \sigma_{\psi i \eta i} \\ \text{Symmetric} & & & \sigma^2_{\eta i} \end{bmatrix} \qquad i = 1, 2, 3$$

Where C_{ij}'s are obtained from table look up.

Test $\rho_i \geq \rho^1_{max}$ if yes $\sigma^2_{\rho i} = 10^6$

if no

$$\sigma^2_{\rho i} = b_{io} + b_{i1} \rho_i^2 + b_{i2} \rho_i^4$$

$$\sigma^2_{\dot{\rho} i} = a_{io} + a_{i1} (1 + b_i \dot{\rho}_i)^2 \rho_i + a_{i2} (1 + b_i \dot{\rho}_i)^2 \rho_i^2 + a_{i3} (1 + b_i \dot{\rho}_i)^4$$

ρ_i and $\dot{\rho}_i$ are the actual values

Test $\rho_i \geq \rho^2_{max}$ if yes

$$\left.\begin{matrix} \sigma^2_{\psi i} \\ \sigma^2_{\eta i} \end{matrix}\right\} = 10^6$$

if no

$\sigma^2_{\psi i}$ and $\sigma^2_{\eta i}$ as in input.

V.2.2 Tracker Observation Matrices

$$
H_{iT1} = \begin{bmatrix}
\frac{\partial \rho_i}{\partial X_1} & \frac{\partial \rho_i}{\partial X_2} & \frac{\partial \rho_i}{\partial X_3} & 0 & 0 & 0 \\
\frac{\partial \dot{\rho}_i}{\partial X_1} & \frac{\partial \dot{\rho}_i}{\partial X_2} & \frac{\partial \dot{\rho}_i}{\partial X_3} & \frac{\partial \dot{\rho}_i}{\partial X_4} & \frac{\partial \dot{\rho}_i}{\partial X_5} & \frac{\partial \dot{\rho}_i}{\partial X_6} \\
\frac{\partial \psi_i}{\partial X_1} & \frac{\partial \psi_i}{\partial X_2} & \frac{\partial \psi_i}{\partial X_3} & 0 & 0 & 0 \\
\frac{\partial \eta_i}{\partial X_1} & \frac{\partial \eta_i}{\partial X_2} & \frac{\partial \eta_i}{\partial X_3} & 0 & 0 & 0
\end{bmatrix} \quad i = 1, 2, 3
$$

$$
H_{iT2} = \begin{bmatrix}
\frac{\partial \rho_i}{\partial X_{iT}} & \frac{\partial \rho_i}{\partial Y_{iT}} & \frac{\partial \rho_i}{\partial Z_{iT}} & 1 & 0 & 0 & 0 \\
\frac{\partial \dot{\rho}_i}{\partial X_{iT}} & \frac{\partial \dot{\rho}_i}{\partial Y_{iT}} & \frac{\partial \dot{\rho}_i}{\partial Z_{iT}} & 0 & 1 & 0 & 0 \\
\frac{\partial \psi_i}{\partial X_{iT}} & \frac{\partial \psi_i}{\partial Y_{iT}} & \frac{\partial \psi_i}{\partial Z_{iT}} & 0 & 0 & 1 & 0 \\
\frac{\partial \eta_i}{\partial X_{iT}} & \frac{\partial \eta_i}{\partial Y_{iT}} & \frac{\partial \eta_i}{\partial Z_{iT}} & 0 & 0 & 0 & 1
\end{bmatrix} \quad i = 1, 2, 3
$$

$$
\frac{\partial \rho_i}{\partial X_1} = \frac{\partial \dot{\rho}_i}{\partial X_4} = - \frac{\partial \rho_i}{\partial X_{iT}} = \frac{X_{ip}(t_k)}{\rho_i\ (t_k)}
$$

$$
\frac{\partial \rho_i}{\partial X_2} = \frac{\partial \dot{\rho}_i}{\partial X_5} = - \frac{\partial \rho_i}{\partial Y_{iT}} = \frac{Y_{ip}(t_k)}{\rho_i\ (t_k)}
$$

$$
\frac{\partial \rho_i}{\partial X_3} = \frac{\partial \dot{\rho}_i}{\partial X_6} = - \frac{\partial \rho_i}{\partial Z_{iT}} = \frac{Z_{ip}(t_k)}{\rho_i\ (t_k)}
$$

$$\frac{\partial \dot{\rho}_i}{\partial X_1} = \frac{1}{\rho_i(t_k)} \left[\dot{X}_{ip}(t_k) - \dot{\rho}_i(t_k) \frac{\partial \rho_i}{\partial X_1} \right]$$

$$\frac{\partial \dot{\rho}_i}{\partial X_2} = \frac{1}{\rho_i(t_k)} \left[\dot{Y}_{ip}(t_k) - \dot{\rho}_i(t_k) \frac{\partial \rho_i}{\partial X_2} \right]$$

$$\frac{\partial \dot{\rho}_i}{\partial X_3} = \frac{1}{\rho_i(t_k)} \left[\dot{Z}_{ip}(t_k) - \dot{\rho}_i(t_k) \frac{\partial \rho_i}{\partial X_3} \right]$$

$$\frac{\partial \dot{\rho}_i}{\partial X_{iT}} = -\frac{\partial \dot{\rho}_i}{\partial X_1} - \frac{\omega Y_{ip}}{\rho_i} \qquad \frac{\partial \dot{\rho}_i}{\partial Y_{iT}} = -\frac{\partial \dot{\rho}_i}{\partial X_2} + \frac{\omega X_{ip}}{\rho_i} \qquad \frac{\partial \dot{\rho}_i}{\partial Z_{iT}} = -\frac{\partial \dot{\rho}_i}{\partial X_3}$$

$$\frac{\partial \psi_i}{\partial X_1} = \frac{1}{\cos \psi_i} \left[\frac{X_{iT}}{r_{iT}\rho_i} - \frac{X_{ip}}{\rho_i^{\,2}} \sin \psi_i \right]$$

$$\frac{\partial \psi_i}{\partial X_2} = \frac{1}{\cos \psi_i} \left[\frac{Y_{iT}}{r_{iT}\rho_i} - \frac{Y_{ip}}{\rho_i^{\,2}} \sin \psi_i \right]$$

$$\frac{\partial \psi_i}{\partial X_3} = \frac{1}{\cos \psi_i} \left[\frac{Z_{iT}}{r_{iT}\rho_i} - \frac{Z_{ip}}{\rho_i^{\,2}} \sin \psi_i \right]$$

$$\frac{\partial \psi_i}{\partial X_{iT}} = \frac{1}{\cos \psi_i} \left[\frac{X_{ip}}{r_{iT}\rho_i} - \frac{X_{iT}}{r_{iT}^{\,2}} \sin \psi_i \right] - \frac{\partial \psi_i}{\partial X_1}$$

$$\frac{\partial \psi_i}{\partial Y_{iT}} = \frac{1}{\cos \psi_i} \left[\frac{Y_{ip}}{r_{iT}\rho_i} - \frac{Y_{iT}}{r_{iT}^{\,2}} \sin \psi_i \right] - \frac{\partial \psi_i}{\partial X_2}$$

$$\frac{\partial \psi_i}{\partial Z_{iT}} = \frac{1}{\cos \psi_i} \left[\frac{Z_{ip}}{r_{iT}\rho_i} - \frac{Z_{iT}}{r_{iT}^{\,2}} \sin \psi_i \right] - \frac{\partial \psi_i}{\partial X_3}$$

$$\frac{\partial \eta_i}{\partial X_1} = \frac{1}{x''_{ip}} \left[+ \sin(\theta_i + \omega t_k) + \frac{y''_{ip}}{\rho_i} \left(\frac{\partial \rho_i}{\partial X_1} \right) - y''_{ip} \left(\frac{\sin \psi_i}{\cos \psi_i} \right) \left(\frac{\partial \psi_i}{\partial X_1} \right) \right]$$

$$\frac{\partial \eta_i}{\partial X_2} = \frac{1}{x''_{ip}} \left[- \cos(\theta_i + \omega t_k) + \frac{y''_{ip}}{\rho_i} \left(\frac{\partial \rho_i}{\partial X_2} \right) - y''_{ip} \left(\frac{\sin \psi_i}{\cos \psi_i} \right) \left(\frac{\partial \psi_i}{\partial X_2} \right) \right]$$

$$\frac{\partial \eta_i}{\partial X_3} = \frac{1}{x''_{ip}} \left[\frac{y''_{ip}}{\rho_i} \left(\frac{\partial \rho_i}{\partial X_3} \right) - y''_{ip} \left(\frac{\sin \psi_i}{\cos \psi_i} \right) \left(\frac{\partial \psi_i}{\partial X_3} \right) \right]$$

$$\frac{\partial \eta_i}{\partial X_{iT}} = \frac{1}{x''_{ip}} \left[- \sin(\theta_i + \omega t_k) - \frac{A Y_{iT}}{X_{iT}^2 + Y_{iT}^2} + \frac{y''_{ip}}{\rho_i} \left(\frac{\partial \rho_i}{\partial X_{iT}} \right) - y''_{ip} \left(\frac{\sin \psi_i}{\cos \psi_i} \right) \left(\frac{\partial \psi_i}{\partial X_{iT}} \right) \right]$$

$$\frac{\partial \eta_i}{\partial Y_{iT}} = \frac{1}{x''_{ip}} \left[\cos(\theta_i + \omega t_k) + \frac{A X_{iT}}{X_{iT}^2 + Y_{iT}^2} + \frac{y''_{ip}}{\rho_i} \left(\frac{\partial \rho_i}{\partial Y_{iT}} \right) - y''_{ip} \left(\frac{\sin \psi_i}{\cos \psi_i} \right) \left(\frac{\partial \psi_i}{\partial Y_{iT}} \right) \right]$$

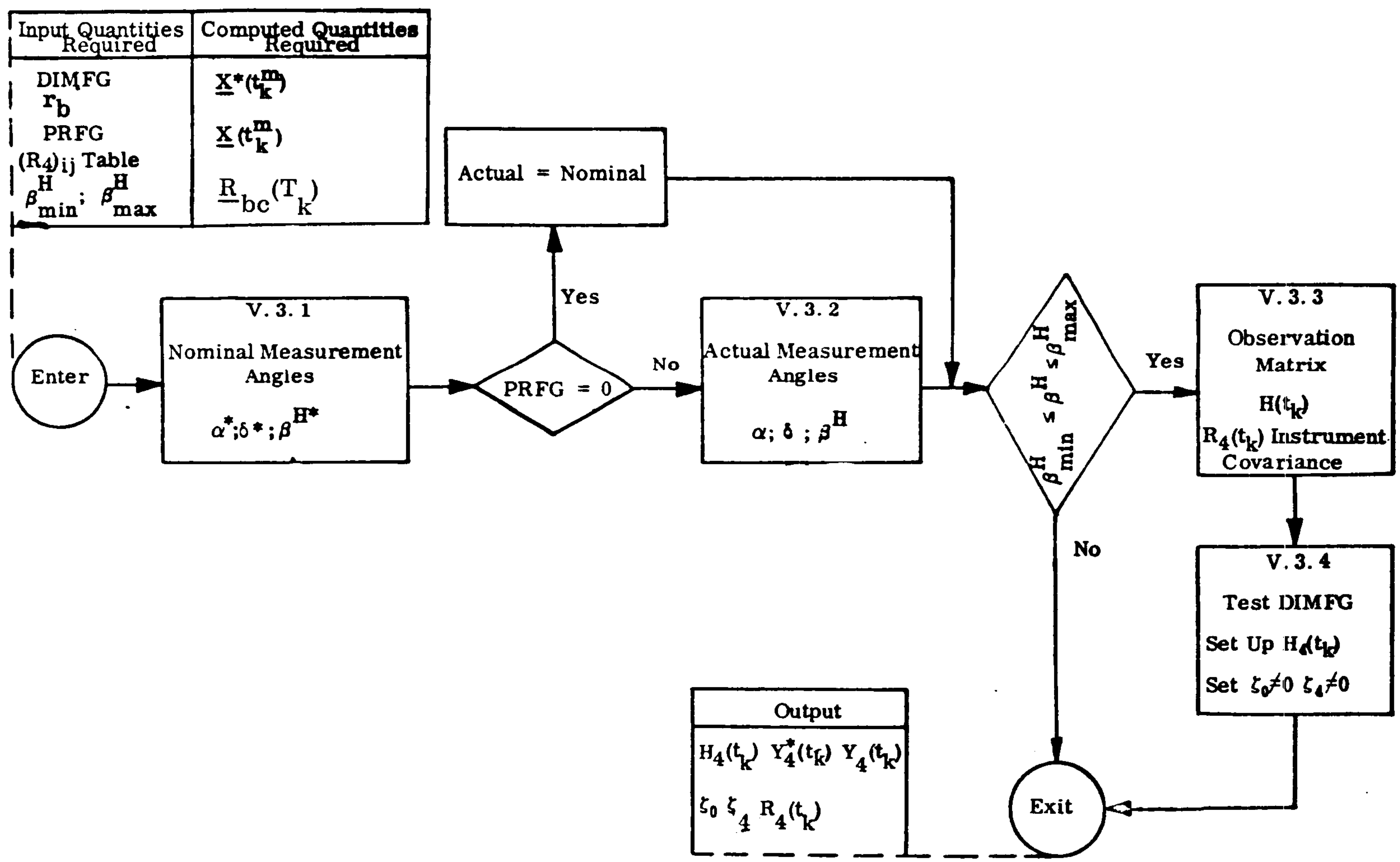
$$\frac{\partial \eta_i}{\partial Z_{iT}} = \frac{1}{x''_{ip}} \left[\frac{y''_{ip}}{\rho_i} \left(\frac{\partial \rho_i}{\partial Z_{iT}} \right) - y''_{ip} \left(\frac{\sin \psi_i}{\cos \psi_i} \right) \left(\frac{\partial \psi_i}{\partial Z_{iT}} \right) \right]$$

where $A = X_{ip} \cos(\theta_i + \omega t_k) + Y_{ip} \sin(\theta_i + \omega t_k)$

These partials are evaluated employing the nominal (*) values.

V.2.3 Set Up of Observation Matrix

The observation matrix form is set up in 3.5.1.3 Dimension Block B.3. It remains here to place the computed submatrice in the proper locations, and set $\zeta_o \neq 0$ $\zeta_i \neq 0 \quad i = 1, 2, 3.$



3.4.5.2.5 Horizon Sensor - Block V.3

V.3.1 Nominal Measurement Angles

Three angles are measured by this instrument:

$$\text{Elevation angle } \alpha^* = -\sin^{-1}\left(\frac{X^*_{pb3}(t_k)}{R^*_{pb}(t_k)}\right)$$

$$\text{Azimuth angle } \delta^* = \begin{cases} \sin^{-1} & \dfrac{X^*_{pb2}(t_k)}{[X^{*2}_{pb1}(t_k) + X^{*2}_{pb2}(t_k)]^{1/2}} \\ \cos^{-1} & \dfrac{X^*_{pb1}(t_k)}{[X^{*2}_{pb1}(t_k) + X^{*2}_{pb2}(t_k)]^{1/2}} \end{cases}$$

$$\text{Subtended angle } \beta^{H*} = \sin^{-1}\left(\frac{r_b}{R^*_{pb}(t_k)}\right)$$

where

$R^*_{pb}(t_k)$ is the magnitude of $\underline{R}^*_{pb}(t_k) = \underline{R}^*(t_k) - \underline{R}_{bc}(T_k)$

r_b = radius of reference body

$\underline{R}_{bc}$ = position of reference body b w.r.t. central body c

$$\underline{R}^*_{pb} = \begin{pmatrix} X^*_{pb1} \\ X^*_{pb2} \\ X^*_{pb3} \end{pmatrix}$$

$$Y^*_4 \overset{Df}{=} \begin{bmatrix} \alpha^* \\ \delta^* \\ \beta^{H*} \end{bmatrix}$$

V.3.2 Actual Measurement Angles

$$Y_4 \overset{Df}{=} \begin{bmatrix} \alpha \\ \delta \\ \beta^H \end{bmatrix}$$

The equations are the same as in V.3.1 except that $\underline{R}(t_k) \rightarrow \underline{R}^*(t_k)$.

Test $\beta^H_{min} \le \beta^H \le \beta^H_{max}$; if yes, continue; if no, exit.

V.3.3 Observation Matrix and Instrument Covariance

$$H(t_k) = \left(\underset{3 \times 3}{H} \quad \underset{3 \times 3}{O} \right)$$

The nontrivial portion of the $H(t_k)$ matrix has dimension (3 x 3). It is represented by the following matrix.

$$\begin{bmatrix} \dfrac{-\sin\alpha^*\cos\delta^*}{R^*_{pb}(t_k)} & -\dfrac{\sin\alpha^*\sin\delta^*}{R^*_{pb}(t_k)} & \dfrac{-[X^{*2}_{pb1}(t_k)+X^{*2}_{pb2}(t_k)]^{1/2}}{R^{*2}_{pb}(t_k)} \\ \dfrac{-X^*_{pb2}(t_k)}{X^{*2}_{pb1}(t_k)+X^{*2}_{pb2}(t_k)} & \dfrac{X^*_{pb1}(t_k)}{X^{*2}_{pb1}(t_k)+X^{*2}_{pb2}(t_k)} & 0 \\ \dfrac{-X^*_{pb1}(t_k)\tan\beta^{H*}}{R^{*2}_{pb}(t_k)} & \dfrac{-X^*_{pb2}(t_k)\tan\beta^{H*}}{R^{*2}_{pb}(t_k)} & \dfrac{-X^*_{pb3}(t_k)\tan\beta^{H*}}{R^{*2}_{pb}(t_k)} \end{bmatrix}$$

Note: $$\tan\beta^{H*} = \frac{r_b}{\sqrt{R^{*2}_{pb}(t_k) - r_b^2}}$$

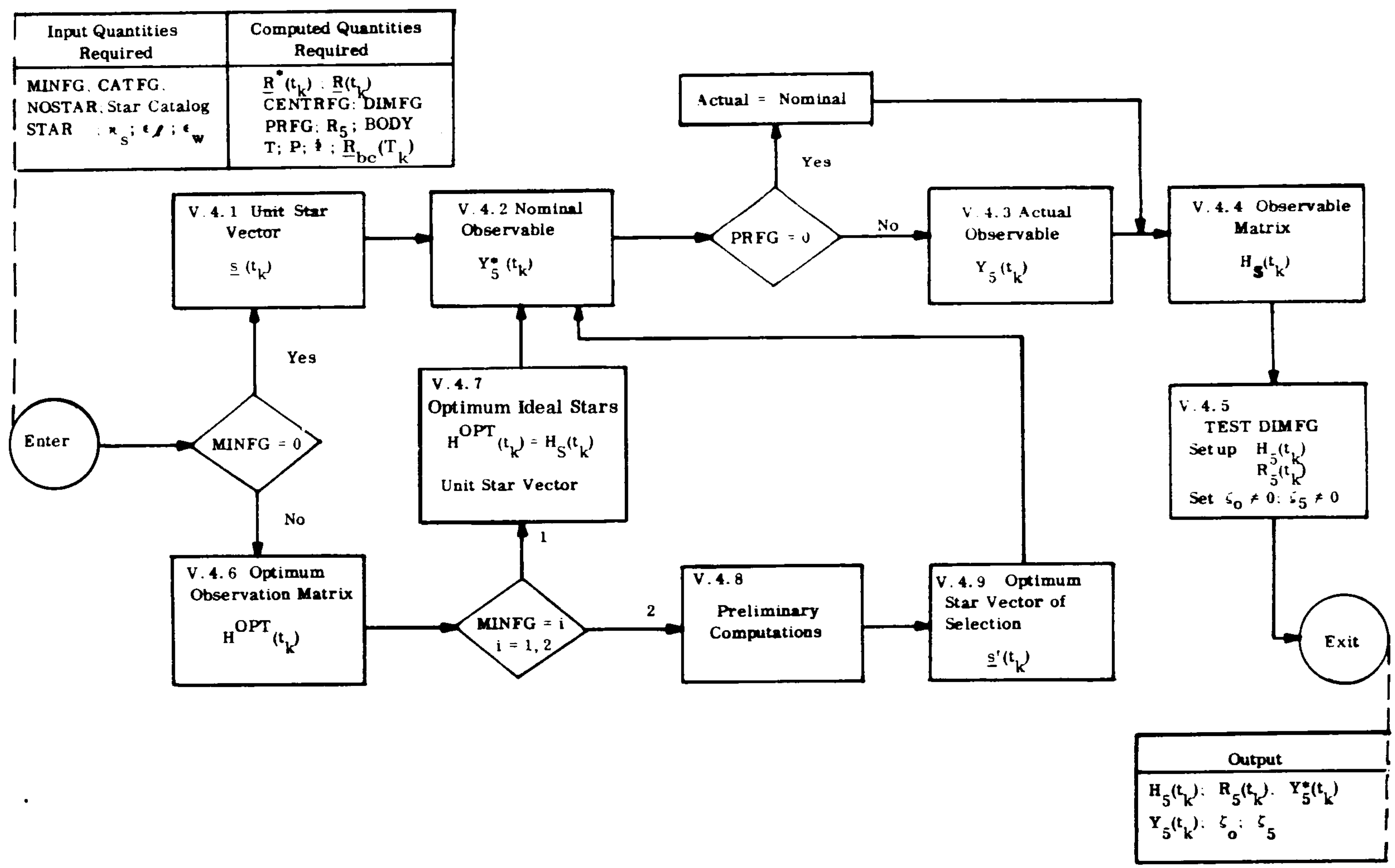
The instrument covariance is given by

$$R_4(t_k) = R_{ij} \qquad \begin{array}{l} i = 1,2,3 \\ j = 1,2,3 \end{array}$$

where the R_{ij}'s are obtained from a table look up as a function of time. Since R_4 is symmetric, only the non-symmetric elements will be part of the input.

V.3.4 Setup of $H_4(t_k)$

The observation matrix form is established in 3.5.1.3 Dimension - Block B.3. It remains here to place the computed submatrices in the proper locations. After $H_4(t_k)$ is set up ζ_o and ζ_4 are set $\neq 0$.



3.4.5.2.6 Space Sextant - Block V.4

V.4.1.1 Star Catalogs

The star catalogs will be preassembled and will have assigned catalog numbers. The particular catalog to be employed will be designated with the input data. The largest number of stars in any catalog as presently envisioned will be 342.

The NOSTAR input quantity will further restrict the specific star catalog, after it is read in, to the number of stars designated by NOSTAR, e.g., NOSTAR = 15 means that only the first 15 stars of the catalog are used.

Each catalog will contain the STAR number j (not necessarily monotonically increasing integers); the right ascension α_j (in radians); and the declination δ_j (in radians) for each star.

The catalogs will be used in two modes designated by the MINFG (input). When MINFG = 0, a specific star j from the employed catalog is to be selected by table look up in the time vs. STAR table. When the MINFG = 2, then the star is selected from optimum observation considerations. In the latter case, the STAR number of the optimum star selected is available for output purposes.

V.4.1.2 Unit Star Vectors

$$\underline{s}_j(t_k) = \begin{bmatrix} s_{j1} \\ s_{j2} \\ s_{j3} \end{bmatrix} = \begin{bmatrix} \cos\alpha_j \;\; \cos\delta_j \\ \sin\alpha_j \;\; \cos\delta_j \\ \sin\delta_j \end{bmatrix}$$

$$j = 1, 2, \ldots, \text{NOSTAR}$$

V.4.2 Nominal Observable

$$Y_5^*(t_k) = \theta^*_{pb}(t_k) = \cos^{-1}\{\underline{s}(t_k) \cdot \underline{e}^*_{pb}(t_k)\} - \kappa_S \beta^{S*}_{pb}(t_k)$$

where

$$\underline{e}^*_{pb}(t_k) = -\frac{\underline{R}^*_{pb}(t_k)}{R^*_{pb}(t_k)}$$

$$\beta^{S*}_{pb}(t_k) = \sin^{-1}\frac{r_b}{R^*_{pb}(t_k)}$$

$$\underline{R}^*_{pb}(t_k) = \underline{R}^*(t_k) - \underline{R}_{bc}(T_k) \qquad \underline{R}_{bb} = 0$$

r_b = radius of reference body $\quad b \overset{Df}{=}$ BODY

$\underline{R}_{bc}(t_k)$ = position of reference body b w.r.t. central body c

$\underline{R}^*_{pb}$ = position of probe w.r.t. reference b.

$$\underline{R}^*(t_k) = \begin{bmatrix} X^*_1(t_k) \\ X^*_2(t_k) \\ X^*_3(t_k) \end{bmatrix}$$

Subscript b $\overset{Df}{=}$ BODY ; b = 0, 1, . . . , 6

V.4.3 Actual Observable

$Y_5(t_k)$; $\underline{e}_{pb}$; β^S_{pb} The equations are the same as in the nominal measurement except that $\underline{R}(t_k) \rightarrow \underline{R}^*(t_k)$.

V.4.4 Observation Matrix

$$H_S(t_k) = [H_{11} \quad H_{12} \quad H_{13} \quad 0 \quad 0 \quad 0]$$

$$H_{11} = \frac{1}{A} [\cos \{\varkappa_S \beta^{S*}_{pb}(t_k)\} s_{j1}(t_k) - \cos \{\theta^*_{pb}(t_k)\} e^*_{pb1}(t_k)]$$

$$H_{12} = \frac{1}{A} [\cos \{\varkappa_S \beta^{S*}_{pb}(t_k)\} s_{j2}(t_k) - \cos \{\theta^*_{pb}(t_k)\} e^*_{pb2}(t_k)]$$

$$H_{13} = \frac{1}{A} [\cos \{\varkappa_S \beta^{S*}_{pb}(t_k)\} s_{j3}(t_k) - \cos \{\theta^*_{pb}(t_k)\} e^*_{pb3}(t_k)]$$

$$A = R^*_{pb}(t_k) \cos \{\varkappa_S \beta^{S*}_{pb}(t_k)\} \sin \{\theta^*_{pb}(t_k) + \varkappa_S \beta^{S*}_{pb}(t_k)\}$$

V.4.5 Setup of $H_5(t_k)$

The observation matrix form is established in 3.5.1.3 Dimension - Block B.3. It remains here to place the computed submatrix in its proper location. After $H_5(t_k)$ is set up ζ_o and ζ_5 are set $\neq 0$.

V.4.6 Optimum Observation Matrix

Let

$$TB^* = \tan \{ \varkappa_S \beta^{S^*}_{pb} (t_k) \}$$

$$co^* = \sqrt{e_2^{*2} + e_3^{*2}}$$

where

$$e^*_{pb}(t_k) = \begin{bmatrix} e_1^* \\ e_2^* \\ e_3^* \end{bmatrix}$$

Compute

$$E^* = (E^*_{ij}) = \begin{bmatrix} -TB^*e_1^* & -TB^*e_2^* & -TB^*e_3^* \\ 0 & e_3^*/co^* & -e_2^*/co^* \\ -co^* & e_1^*\, e_2^*/co^* & e_1^*\, e_3^*/co^* \end{bmatrix}$$

Partition the 6 x 6 matrix $P(t_k)$ into 3 x 3 submatrices as follows

$$P(t_k) = \begin{bmatrix} P_{11}(t_k) & P_{12}(t_k) \\ P^T_{12}(t_k) & P_{22}(t_k) \end{bmatrix}$$

Compute

$$A^* = E^* [P_{11}(t_k)\ P_{12}(t_k)]\ \Phi^T(t_A, t_k)\ T^T(t_A)\ W^T$$

$$W\ T(t_A)\ \Phi(t_A, t_k) \begin{bmatrix} P_{11}(t_k) \\ \\ P_{12}^T(t_k) \end{bmatrix} \qquad E^{*T} = (A^*_{ij})$$

where

$$W = \begin{bmatrix} W_1 & 0 & 0 & 0 \\ 0 & W_2 & 0 & 0 \\ 0 & 0 & W_3 & 0 \\ 0 & 0 & 0 & W_4 \end{bmatrix}$$

$$B^* = (B^*_{ij}) = E^* P_{11}(t_k) E^{*T}$$

Add the quantity $[R^*_{pb}(t_k)]^2 R_5$ to B^*_{11}

Find $\alpha^{OPT}(t_k)$, the angle between 0 and 2π which maximizes

$$Q(\alpha) = \frac{[1, \cos\alpha, \sin\alpha]\ A^* \begin{bmatrix} 1 \\ \cos\alpha \\ \sin\alpha \end{bmatrix}}{[1, \cos\alpha, \sin\alpha]\ B^* \begin{bmatrix} 1 \\ \cos\alpha \\ \sin\alpha \end{bmatrix}}$$

In order to find $\alpha^{OPT}(t_k)$ the following scheme is used.

$Q(\alpha)$ is evaluated at intervals of $\Delta\alpha = 5^o$ until 3 successive evaluations of $Q(\alpha)$ at α_{i-1}, α_i, and α_{i+1} satisfy the following conditions.

$$Q(\alpha_{i-1}) < Q(\alpha_i) \geq Q(\alpha_{i+1})$$

When these conditions are satisfied it is assumed that a relative maximum lies in the interval $(\alpha_{i-1}, \alpha_{i+1})$. If α_i were selected as the value which maximizes $Q(\alpha)$ the error in α would be $\leq 5^o$. To reduce this error by half $Q'(\alpha)$ is evaluated at α_{i-1}, α_i, and α_{i+1}. The value of α in question in then between the two α's for which $Q'(\alpha)$ changes sign, and if the midpoint of this interval is taken the error in α is bounded by 2.5^o. This "halving" procedure is repeated for a total of 4 times. The error in α is then bounded by $.3125^o$. The values of α and $Q(\alpha)$ which are finally selected are stored as relative maxima and when the entire interval $[0, 2\pi]$ has been searched the global maximum is picked. In the case where two relative maxima are equal, the one with the lesser value of α is chosen.

Note that if there is more than one maximum within the 10^o interval that the value of α selected can be as much as 10^o off. However, experience indicates that the maxima are not close together.

Finally, set

$$H_{1j}^{OPT} = \frac{1}{R^*_{pb}(t_k)} \left\{ E^*_{1j} + E^*_{2j} \cos \alpha^{OPT}(t_k) + E^*_{3j} \sin \alpha^{OPT}(t_k) \right\}$$

$$j = 1,2,3$$

$$H^{OPT}(t_k) = [H_{11}^{OPT} \; H_{12}^{OPT} \; H_{13}^{OPT} \quad 0 \quad 0 \quad 0]$$

V.4.7 "Optimum" Ideal Stars

Let $H_s(t_k) \overset{Df}{=} H^{OPT}(t_k)$

and the unit star is defined as

$$\underline{s}''(t_k) = \begin{bmatrix} s_1'' \\ s_2'' \\ s_3'' \end{bmatrix}$$

$$s_1'' = \frac{H_{11}^{OPT}}{\sqrt{(H_{11}^{OPT})^2 + (H_{12}^{OPT})^2 + (H_{13}^{OPT})^2}} = \frac{H_{11}^{OPT}}{h^{OPT}}$$

$$s''_2 = \frac{H_{12}^{OPT}}{h^{OPT}}$$

$$s''_3 = \frac{H_{13}^{OPT}}{h^{OPT}}$$

Set

$$\underline{s}(t_k) = \underline{s}''(t_k)$$

V.4.8 Preliminary Computations

$$\underline{e}^*_{p\ell} = -\left(\frac{\underline{R}^*_{p\ell}}{R_{p\ell}}\right) \quad ; \quad T_k = T_L - C_T(t_k - t_o)$$

$$\beta^*_{p\ell} = \sin^{-1}\left(\frac{r_\ell}{R^*_{p\ell}}\right)$$

$$\underline{R}^*_{p\ell} = \underline{R}^* - \underline{R}_{\ell c}$$

$$\beta_{pb} = \sin^{-1}\left(\frac{r_b}{R^*_{pb}}\right) \text{ ; subscript b } \overset{Df}{=} \text{ BODY}$$

$\underline{R}^*_{p\ell}$ is the position of the probe w.r.t. body ℓ

$\underline{R}^*_{\ell c}$ is the position of the body ℓ w.r.t. the central body $\hat{c}$ = CENTRFG
$\ell = 0, 1, 2, 3, \ldots, 6$

$\underline{R}^*_{pb}$ is the position of the probe w.r.t. reference body b. When the reference body is also the central body, b = c.

r_b radius of reference body.

$\epsilon_w = \pi/2$ unless over-ridden by an input value.

The central body c = CENTRFG is a function of the conic m and is part of the input. The CENTRFG is set in either Block II or III. The reference body b (b = 0,1,2,...,6) will be inputed in terms of a table look up in time.

V.4.9 Optimum Star Vector Selection

Use V.4.1 to compute unit star vectors $\underline{s}_j(t_k)$ from appropriate star catalog. Select unit star vector $\underline{s}_j$ such that it satisfies the following constraints:

1. $\underline{s}_j \cdot \underline{h}_{OPT} > 0$

2. $\cos^{-1}(\underline{s}_j \cdot \underline{e}^*_{pb}) < (\beta_{pb} + \epsilon_w)$

3. $\cos^{-1}(\underline{s}_j \cdot \underline{e}^*_{p\ell} > (\beta_{p\ell} + \epsilon_\ell)$

4. $\delta\alpha = \cos^{-1} \dfrac{(\underline{s}_j \times \underline{e}^*_{pb}) \cdot (\underline{h}_{OPT} \times \underline{e}^*_{pb})}{|\underline{s}_j \times \underline{e}^*_{pb}|\ |\underline{h}_{OPT} \times \underline{e}^*_{pb}|} = \text{Minimum}$

Designate the unit star vector that satisfies the four constraints above by $\underline{s}'(t_k)$

Set

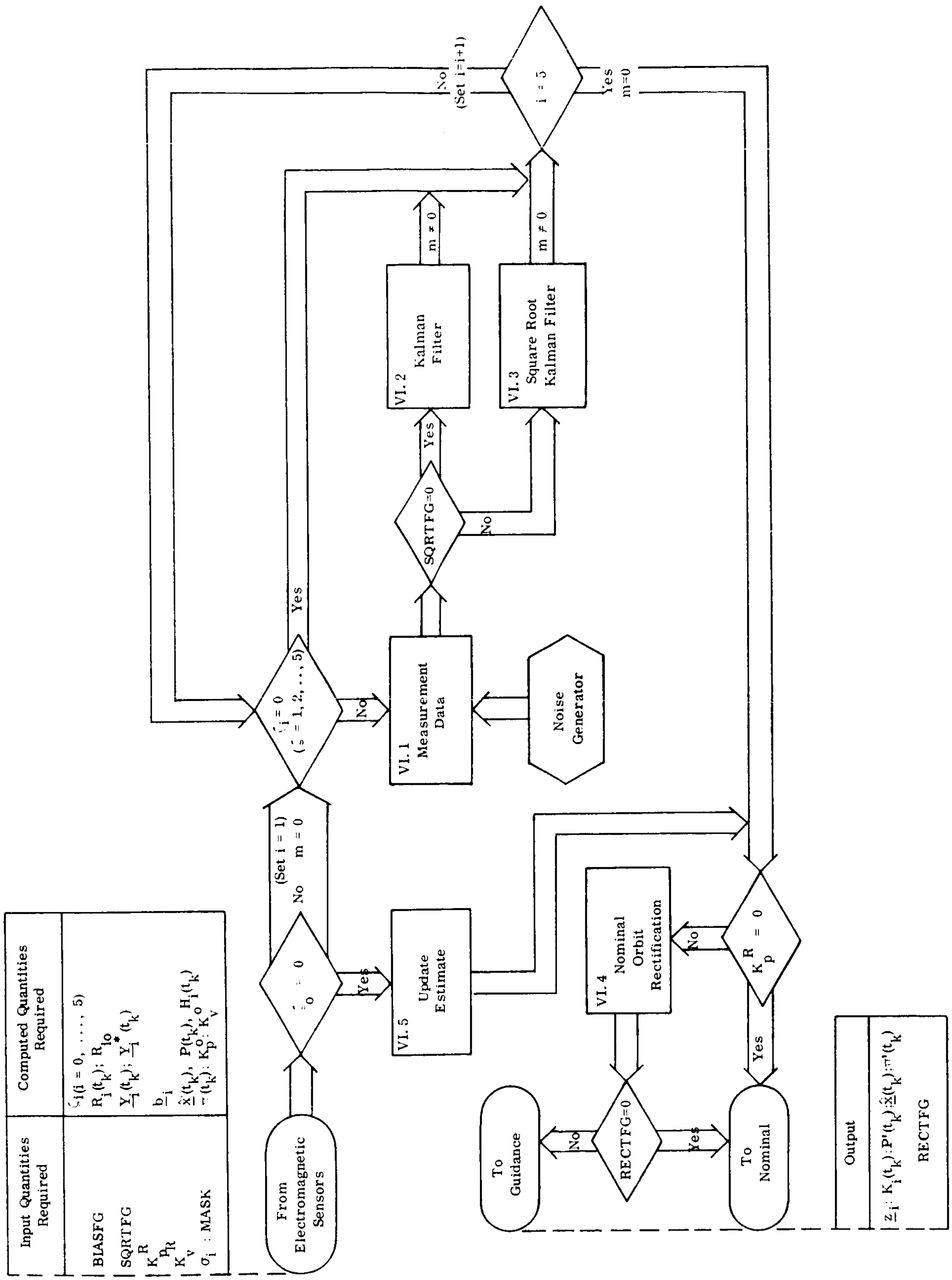
$\underline{s}'(t_k) \overset{Df}{=} \underline{s}(t_k)$; then compute the nominal and actual observables and the observation matrix using $\underline{s}(t_k)$.

where

$$\underline{e}^*_{pb}(t_k) = - \frac{\underline{R}^*_{pb}(t_k)}{R^*_{pb}(t_k)} \quad \text{unit vector to reference body b.}$$

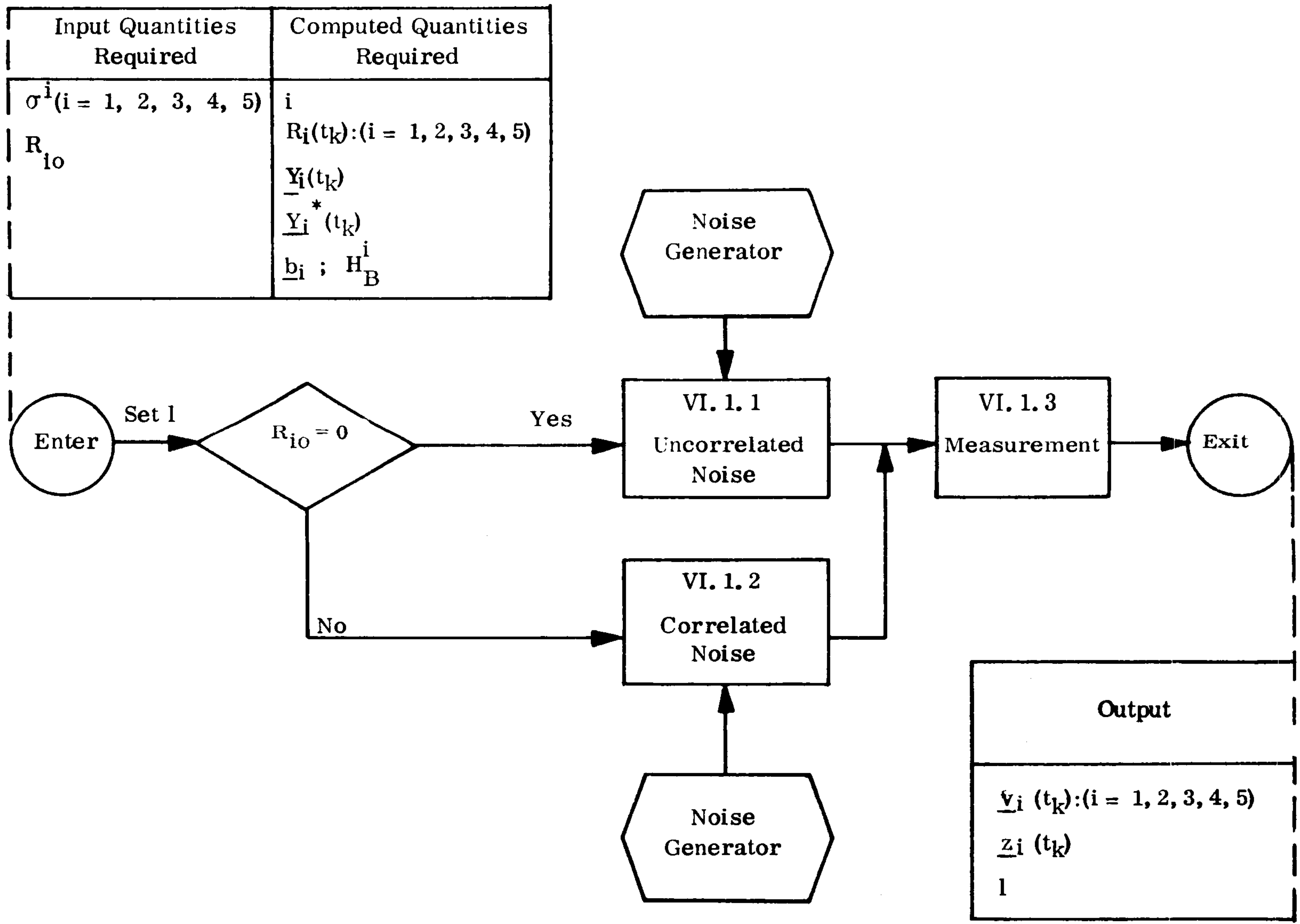
$$\underline{h}(t_k)_{OPT} = \begin{bmatrix} H^{OPT}_{11} \\ H^{OPT}_{12} \\ H^{OPT}_{13} \end{bmatrix}$$

3.4.6 Navigation - Block VI



3.4.6.1 Level II Flow Chart - Navigation

3.4.6.2 Detailed Flow Charts and Equations



3.4.6.2.1 Measurement Data - Block VI. 1

Set ℓ:

If

$$i = \begin{cases} 5 \\ 4 \\ 1,\ 2,\ 3 \end{cases}, \Longrightarrow \ell = \begin{cases} 1 \\ 3 \\ 4 \end{cases}$$

VI. 1. 1 Uncorrelated Noise

Generate ℓ gaussian random numbers with mean zero and variance determined by the diagonal elements of the $R_i(t_k)$ matrix.

$$\text{variance of } j^{th} \text{ random number } = (1 + \sigma^i)\, R^i_{jj}(t_k)\ ,\quad j = 1, 2, \ldots,$$

These random numbers shall form the vector

$$\underline{\eta}_i(t_k) = \begin{bmatrix} \eta^i_1(t_k) \\ \cdot \\ \cdot \\ \cdot \\ \eta^i_\ell(t_k) \end{bmatrix}$$

Let

$$\underline{v}_i(t_k) = \underline{\eta}_i(t_k)$$

VI. 1. 2 Correlated Noise - Triangularization Algorithm

Form a lower triangular matrix $T_{R_i}(t_k)$ according to the following procedure

Suppose

$$T_{R_i}(t_k) = \begin{bmatrix} 1 & & \ldots & 0 \\ \alpha_{21} & 1 & & \vdots \\ \vdots & \alpha_{32} & & 0 \\ \alpha_{\ell 1} & \alpha_{\ell 2} & \ldots & 1 \end{bmatrix}$$

Compute

1. variance (η_i) $= E[\eta_1^2] = E[v_1^2] \underline{\underline{Df}} R^i_{11}(t_k)$

$$\text{then} \quad \alpha_{21} = \frac{E[v_1 v_2]}{E[\eta_1^2]} \underline{\underline{Df}} \frac{R^i_{12}(t_k)}{R^i_{11}(t_k)}$$

$$\alpha_{31} = \frac{E[v_1 v_3]}{E[\eta_1^2]} \underline{\underline{Df}} \frac{R^i_{13}(t_k)}{R^i_{11}(t_k)}$$

$$\alpha_{41} = \frac{E[v_1 v_4]}{E[\eta_1^2]} \underline{\underline{Df}} \frac{R^i_{14}(t_k)}{R^i_{11}(t_k)}$$

2. variance (η_2^i) $= E[\eta_2^2] = E[v_2^2] - \alpha_{21}^2 E[\eta_1^2] \underline{\underline{Df}} R^i_{22}(t_k) - \alpha_{21} R^i_{11}(t_k)$

$$\text{then} \quad \alpha_{32} = \frac{E[v_2 v_3] - \alpha_{21}\alpha_{31} E[\eta_1^2]}{E(\eta_2^2)} \underline{\underline{Df}} \frac{R^i_{23}(t_k) - \alpha_{21}\alpha_{31} R^i_{11}(t_k)}{R^i_{22}(t_k) - \alpha_{21} R^i_{11}(t_k)}$$

$$\alpha_{42} = \frac{E[v_2 v_4] - \alpha_{21}\alpha_{41} E[\eta_1^2]}{E[\eta_2^2]} \underline{\underline{Df}} \frac{R^i_{24}(t_k) - \alpha_{21}\alpha_{41} R^i_{11}(t_k)}{R^i_{22}(t_k) - \alpha_{21} R^i_{11}(t_k)}$$

3. variance (η_3^i) $= E[\eta_3^2] = E[v_3^2] - \alpha_{31}^2 E[\eta_1^2] - \alpha_{32}^2 E[\eta_2^2]$

$$\underline{\underline{Df}} \quad R^i_{33}(t_k) - \alpha_{31}^2 R^i_{11}(t_k) - \alpha_{32}^2 R^i_{22}(t_k)$$

$$\text{then} \quad \alpha_{43} = \frac{E[v_3 v_4] - \alpha_{31}\alpha_{41} E[\eta_1^2] - \alpha_{32}\alpha_{42} E[\eta_2^2]}{E[\eta_3^2]}$$

4. Proceeding inductively, we have in general

$$\text{variance } (\eta_j^i) = E[\eta_i^2] = E[v_j^2] - \alpha_{j1}^2 E[\eta_1^2] - \alpha_{j2}^2 E[\eta_2^2]$$

$$- \dots - \alpha_{jj-1}^2 E[\eta_{j-1}^2]$$

$$\alpha_{\ell j} = \frac{E[v_j v_\ell] - \alpha_{j1}\alpha_{\ell -1} E[\eta_1^2] - \alpha_{j2}\alpha_{\ell 2} E[\eta_2^2]}{E[\eta_j^2]}$$

$$\frac{- \dots - \alpha_{jj-1}\alpha_{\ell j-1} E[\eta_{j-1}^2]}{E[\eta_j^2]}$$

Note that if $E[\eta_j^2] = 0$ for any j, then the corresponding $\alpha_{\ell j}$ is set equal to zero.

Now generate ℓ gaussian random numbers with mean zero and variance $(1+\sigma^i)\, E[\eta_j^2]$ $j = 1, \dots, \ell$. These random numbers shall form the vector

$$\underline{\eta}_i(t_k) \underline{\underline{Df}} \begin{bmatrix} \eta_1^i(t_k) \\ \eta_2^i(t_k) \\ \vdots \\ \eta_\ell^i(t_k) \end{bmatrix}$$

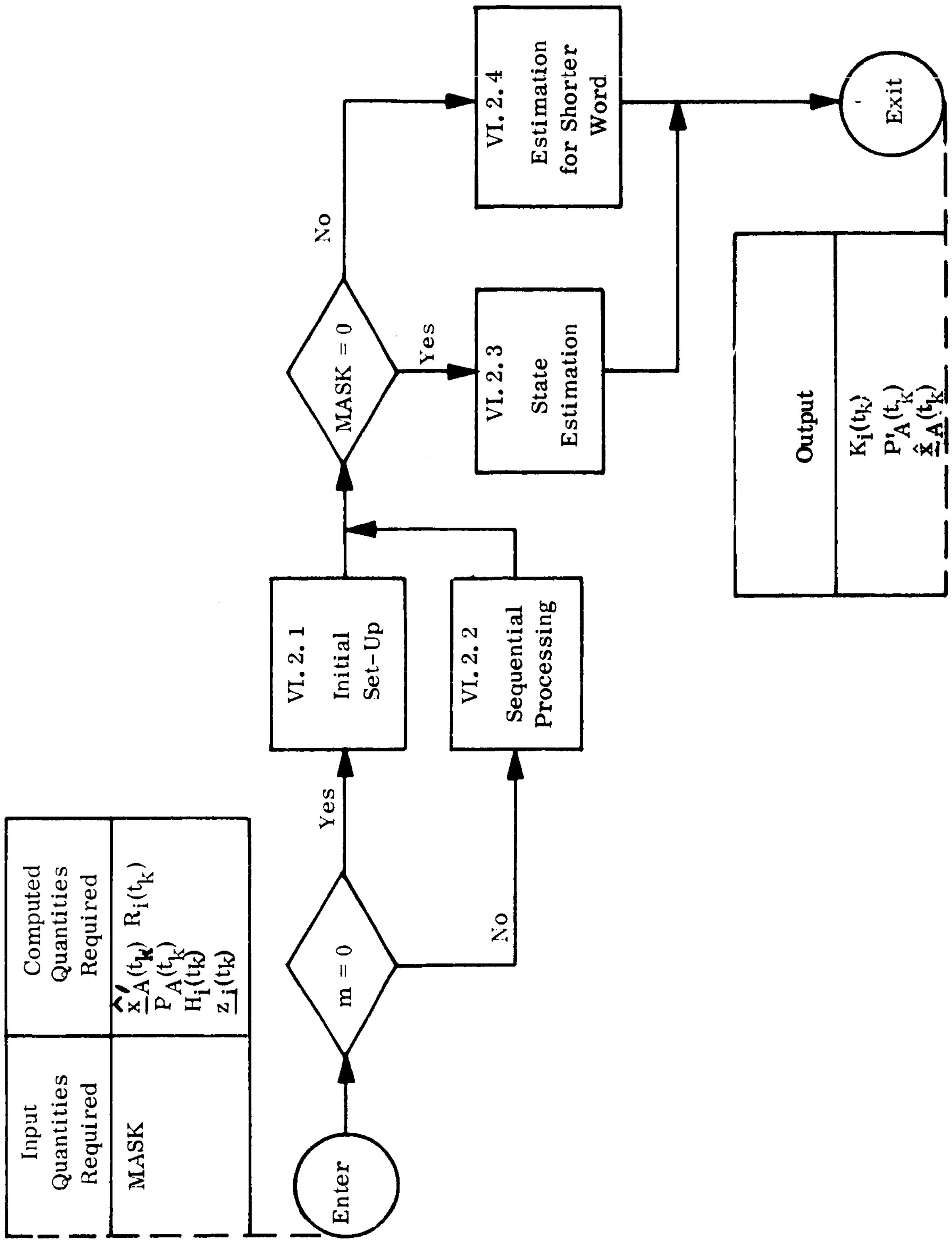
then

$$\underline{v}_i(t_k) = T_{Ri}(t_k)\, \underline{\eta}_i(t_k)$$

VI. 1. 3 Measurements

$$\underline{y}_i(t_k) = \underline{Y}_i(t_k) - \underline{Y}_i^*(t_k)$$

$$\underline{z}_i(t_k) = \underline{y}_i(t_k) + H_B^i \underline{b}_i + \underline{v}_i(t_k)$$



3.4.6.2.2 Kalman Filter - Block VI. 2

VI. 2. 1 Initial Set-Up

Let

$$\hat{\underline{x}}''(t_k) = \hat{\underline{x}}'_A(t_k)$$

$$P''(t_k) = P_A(t_k)$$

Set

$$m \neq 0$$

VI. 2. 2 Sequential Processing

Let

$$\hat{\underline{x}}''(t_k) = \hat{\underline{x}}_A(t_k)$$

$$P''(t_k) = P'_A(t_k)$$

VI. 2. 3 State Estimation

$$K_i(t_k) = P''(t_k)\, H_i^T(t_k)\, [H_i(t_k)\, P''(t_k)\, H_i^T(t_k) + R_i(t_k)]^{-1}$$

$$P'_A(t_k) = P''(t_k) - K_i(t_k)\, H_i(t_k)\, P''(t_k)$$

$$\hat{\underline{x}}_A(t_k) = \hat{\underline{x}}''(t_k) + K_i(t_k)\, [\underline{z}_i(t_k) - H_i(t_k)\, \hat{\underline{x}}''(t_k)]$$

VI. 2. 4 Estimation for Shorter Word

Set last MASK bits of $P''(t_k)$ (i.e., every element) equal to zero.

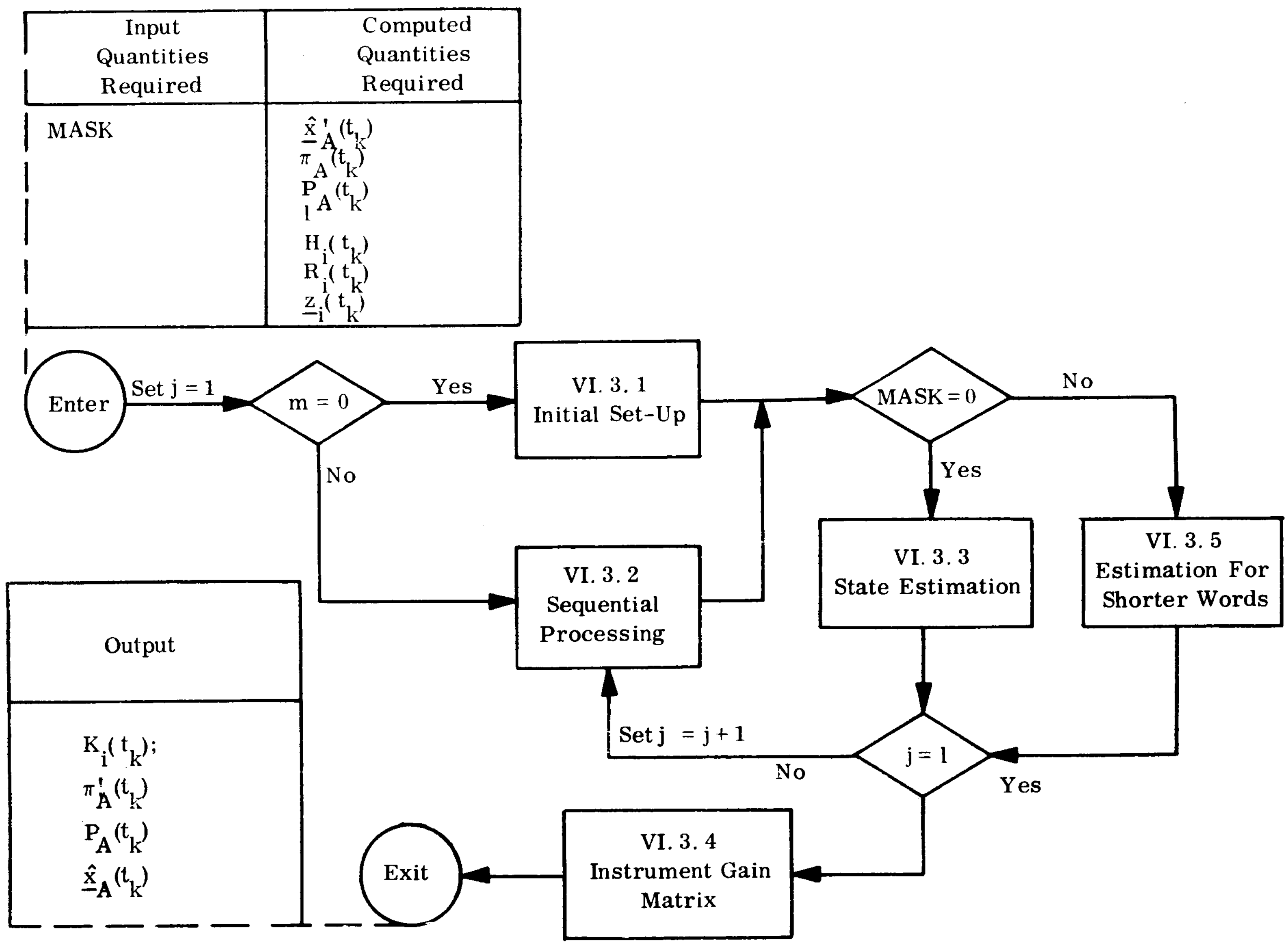
$$K_i(t_k) = P''(t_k)\, H_i^T(t_k)\, [H_i(t_k)\, P''(t_k)\, H_i^T(t_k) + R_i(t_k)]^{-1}$$

Set last MASK bits of $K_i(t_k)$ (i.e., every element) equal to zero

$$P'_A(t_k) = P''(t_k) - K_i(t_k)\, H_i(t_k)\, P''(t_k)$$

$$\hat{\underline{x}}_A(t_k) = \hat{\underline{x}}''(t_k) + K_i(t_k)\, [\underline{z}_i(t_k) - H_i(t_k)\, \hat{\underline{x}}'(t_k)]$$

Set last MASK bits of $\hat{\underline{x}}_A(t_k)$ (i.e., every component) equal to zero.



3.4.6.2.3 Square-Root Kalman Filter - Block VI. 3

VI. 3. 1 Initial Setup

Let

$$\hat{\underline{x}}''(t_k) = \hat{\underline{x}}'_A(t_k)$$

$$\Pi''(t_k) = \Pi_A(t_k)$$

$$P''(t_k) = P_A(t_k)$$

Set

$$m \neq 0$$

VI. 3. 2 Sequential Processing

Let

$$\hat{\underline{x}}''(t_k) = \hat{\underline{x}}_A(t_k)$$

$$\Pi''(t_k) = \Pi'_A(t_k)$$

$$P''(t_k) = P_A(t_k)$$

VI. 3. 3 State Estimation

Let H^i_j represent the j^{th} row of the matrix $H_i(t_k)$, (i.e., $H_i(t_k)$ is the observation matrix for the i^{th} observation option $i = 1, 2, \ldots, 5$)

$$J^i_j(t_k) = H^i_j(t_k)\, \Pi''(t_k)$$

$$r^i_j(t_k) = J^i_j(t_k)\, J^{i^T}_j(t_k) + R^i_j(t_k)$$

Note that $R^i_j(t_k)$ represents the j^{th} diagonal element of $R_i(t_k)$

$$K^i_j(t_k) = \frac{1}{r^i_j(t_k)}\, P''(t_k)\, H^{i^T}_j(t_k)$$

$$a_j^i(t_k) = \frac{1}{J_j^i(t_k)\, J_j^{i^T}(t_k)} \left[1 - \sqrt{\frac{R_j^i(t_k)}{r_j^i(t_k)}} \right]$$

$$\Pi'_A(t_k) = \Pi''(t_k)\, [I - a_j^i(t_k)\, J_j^{i^T}(t_k)\, J_j^i(t_k)]$$

$$P'_A(t_k) = \Pi'(t_k)\, \Pi'^T(t_k)$$

$$\hat{\underline{x}}_A(t_k) = \hat{\underline{x}}''(t_k) + K_j^i(t_k)\, [z_j^i(t_k) - H_j^i(t_k)\, \hat{\underline{x}}''(t_k)]$$

where $z_j^i(t_k)$ is the j^{th} component of the vector $\underline{z}_i(t_k)$.

Important: Save each K_j^i for use in VI. 3. 4.

VI. 3. 4 Instrument Gain Matrix

Form

$$A_j^i = (I - K_j^i H_j^i) \quad , \quad j = 2, \ldots, \ell$$

The $(n \times \ell)$ gain matrix is formed as

$$K_i(t_k) = [\, \underbrace{A_\ell^i A_{\ell-1}^i \ldots A_2^i K_1^i}_{n \times 1} \quad \underbrace{A_\ell^i A_{\ell-1}^i \ldots A_3^i K_2^i}_{n \times 1} \quad \ldots\ldots \quad \underbrace{A_\ell^i K_{\ell-1}^i}_{n \times 1} \quad \underbrace{K_\ell^i}_{n \times 1} \,]$$

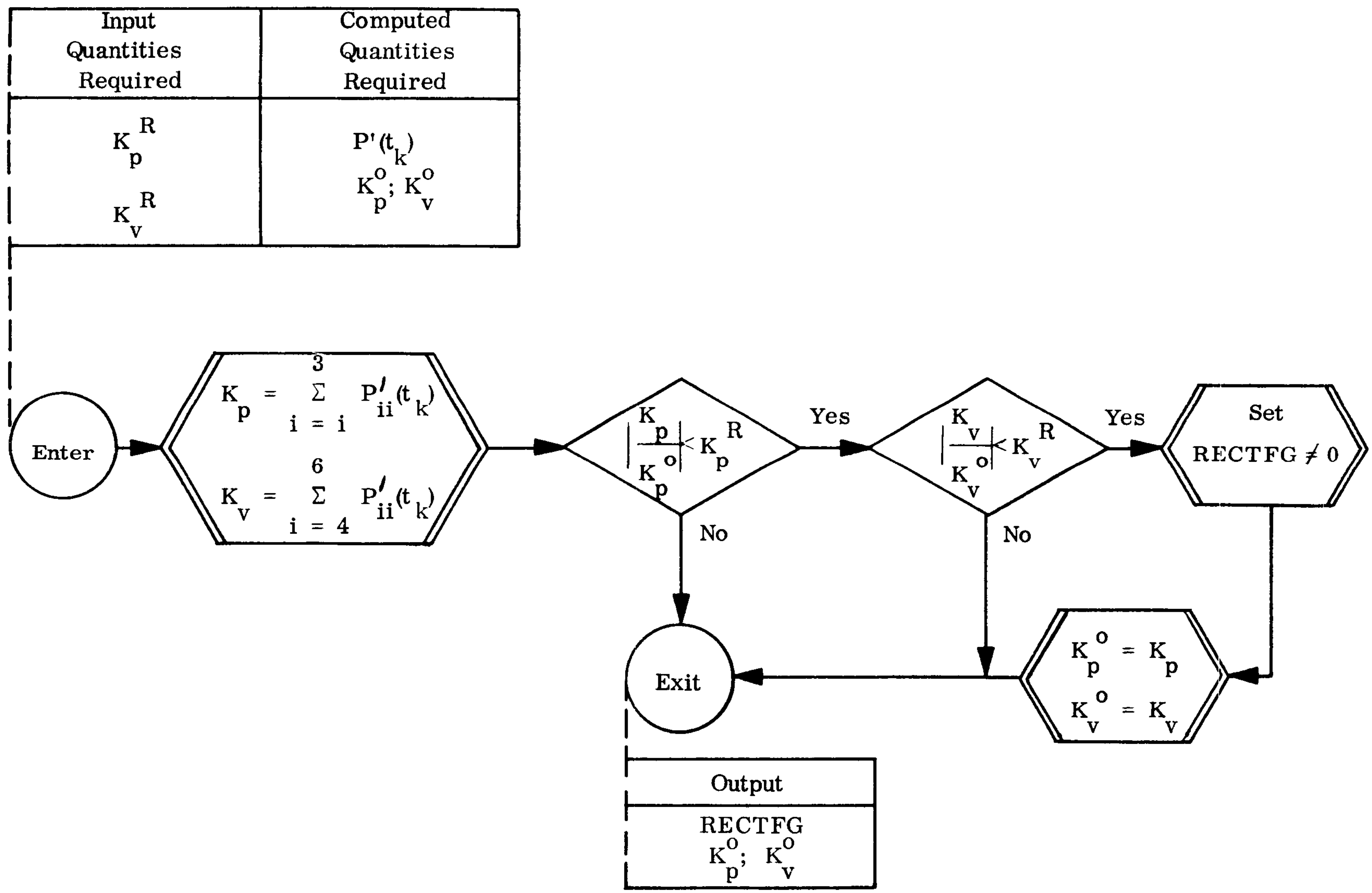
VI. 3. 5 Estimation for Shorter Words

Same equations as described in VI. 3. 3 except the following operations must be performed.

1. Set the last MASK bits of every element of $\Pi''(t_k)$ and $P''(t_k)$ equal to zero.

2. After $K_j^i(t_k)$ is computed, set the last MASK bits of every element equal to zero.

3. After $\hat{\underline{x}}_A(t_k)$ is computed, set the last MASK bits of every component equal to zero.



3.4.6.2.4 Nominal Orbit Rectification - Block VI. 4

3.4.6.2.5 Update Estimate - Block VI.

Set

$$K_i(t_k) \equiv 0 \quad , \quad i = 1, 2, \ldots, 5$$

$$\hat{\underline{x}}_A(t_k) = \hat{\underline{x}}'_A(t_k)$$

$$P'_A(t_k) = P_A(t_k)$$

$$\hat{\underline{z}}_i(t_k) \equiv 0 \quad , \quad i = 1, 2, \ldots, 5$$

and, if necessary,

$$\pi'_A(t_k) = \pi_A(t_k)$$

If MASK $\neq$ 0; set last MASK bits of $P'_A(t_k)$ and $\hat{\underline{x}}_A(t_k)$ equal to zero.

4.0 USER'S GUIDE: PROGRAM 284

4.1 INTRODUCTION

Program 284 is organized in such a manner that the desired program complexity and aiding instrument combination can be simulated. The specific complexity and/or combination is established through the different input flags. Initially, all the flags are set equal to zero, thus, in order to simulate a certain block, the flags and input quantities pertaining to that block will have to be input. Naturally, in order to simulate subsequent blocks which require information from preceding blocks, the preceding blocks have to be simulated also.

The instrument flags in conjunction with the bias flag establish the overall dimension of the system configuration to be simulated. The largest possible system configuration consists of a (31 x 1) augmented state. Therefore, the largest possible matrices that have to be handled by the program are (31 x 31). However, due to sequential processing of the data, the largest matrix that has to be inverted is only (4 x 4). The (31 x 1) augmented state is set up in this simulation as follows:

$\underline{x}$ (6 x 1)	position and velocity components
$\underline{b}_1$ (7 x 1)	1st ground tracker biases
$\underline{b}_2$ (7 x 1)	2nd ground tracker biases
$\underline{b}_3$ (7 x 1)	3rd ground tracker biases
$\underline{b}_4$ (3 x 1)	horizon sensor biases
$\underline{b}_5$ (1 x 1)	space sextant bias

The tracker biases $\underline{b}_i$ (i = 1, 2, 3) further take on the form

$$\underline{b}_i = \begin{cases} \left.\begin{array}{l} x_{1T} \\ x_{2T} \\ x_{3T} \end{array}\right\} & \text{the three components of the tracker location errors} \\ \rho & \text{tracker range bias} \\ \dot{\rho} & \text{tracker range rate bias} \\ \psi & \text{tracker elevation angle bias} \\ \eta & \text{tracker azimuth angle bias} \end{cases}$$

The horizon sensor biases are ordered as follows

$$\underline{b}_4 = \begin{cases} \left.\begin{array}{l} \alpha \\ \delta \end{array}\right\} & \text{local vertical angle biases} \\ \beta^H & \text{planet subtended angle bias} \end{cases}$$

The smallest possible state configuration consists of a (6 x 1) unaugmented state. In this case, the bias flag is equal to zero (i.e., BSFG ≡ 0). The bias flag establishes the desirability of simulating the aiding instrument biases by augmenting the state vector. The BSFG can have only two values—either zero or one. When the BSFG = 1. the program is set up to simulate a system configuration containing the biases of all the specified aiding instruments. There is no direct capability (e.g., variable bias flags) of simulating the biases of a subset of instruments from a particular system configuration. However, one can simulate this situation by properly choosing through input the initial covariances of the instrument biases. For example, suppose that the system configuration includes a horizon sensor and a space sextant as the aiding instruments and it is desirable to include in the simulation only the horizon sensor biases. One would set the BSFG = 1. This will assure that the system configuration dimension is set up properly (see B.3.2 of the program definition). The space sextant bias can then be eliminated by simply setting it equal to zero, (i.e., setting the initial covariance matrix $B_5 \equiv 0$). Other system configuration situations can be treated in a similar way.

The program is arranged so that one can use any set of consistent input units. Furthermore, through a proper choice of conversion factors (an input), one can select different "internal" computational and output-print units. The sets of units selected for input quantities, internal quantities, and output quantities have to be consistent. There is no provision for converting only a few quantities. All the quantities are converted to the same units. It goes without saying that the quantities which are dimensionless, expressed in degrees and/or radians, are not affected by the conversion factors.

4.2 INPUT SHEETS AND GENERAL COMMENTS

The input sheets are closely related to the computational blocks of the program definition. All the quantities have symbols above the addresses indicated on the sheets. For a more detailed definition, one is referred to the program definition. The input units are also given (where applicable) next to the symbols to facilitate easier input generation. The address to the left of each line (card) is associated with the first quantity on that line. Subsequent quantities, separated by commas, have monotonically increasing integer addresses. If it is undesirable to input certain quantities in the middle of a card, it is necessary to input zeros for those quantities so as to preserve the locations properly. For example, suppose that it is desirable to input the radius of the Earth and Venus but not the Moon and Sun; then that particular entry would look like

		RL(0)		RL(1)		RL(2)		RL(3)
4 2 4 0 3	DEC	.6378165E4	,	0	,	0	,	.62E4

The input quantities can be input in either fixed or floating point. In the program, the necessary flags and/or quantities are then set up properly.

4.2.1 Setup and Handling of the Input Tables

There are several input tables with time as the independent variable. Most of these tables can have a maximum of fifty (50) entries. Generally the input sheets do not have the total number of entry locations; however, one could always use blank load sheets to extend the tables if necessary. The main consideration in extending or setting up of the tables is to make sure that subsequent entries have proper addresses.

There are two ways of entering the addresses of subsequent cards. The first is to simply increment the address by the number of entries in the previous card. For example, if the first card has the address of 5 1 5 5 and has 4 entries in it, the second card address will be 5 1 5 9, e.g.,

		TIME	C1I	C2I	C3I
5 1 5 5	DEC	,	,	,	
5 1 5 9	DEC	,	,	,	
5 1 6 3	DEC	,	,	,	

The second way is to simply place a 1 in column 6 of the subsequent cards, e.g.,

			TIME	C1I	C2I	C3I
5 1 5 5		DEC	,	,	,	
	1	DEC	,	,	,	
	1	DEC	,	,	,	

The second method is normally easier to input. However, in the second method, one would have to input all the quantities on a card to preserve the locations properly. Thus, if a certain quantity is not used, a zero must be put in its place.

The independent variable of time is set up in such a manner in these tables that the quantities are used up to the time (t_i) specified, e.g.,

$t \le t_1$ use quantities associated with t_1

$t \le t_2$ use quantities associated with t_2

$t \le t_3$ etc.

4.2.1.1 The Observation Schedule Table

The only time table which is set up somewhat differently is the observation schedule. In this table, in place of the time, one has an entry specifying the number of observations (No. of OBS.). The second entry is the observation interval (OBS. DELTA). The third entry is the print flag (PRINFG). The PRINFG is used to establish the desirability of printing the output in the interval specified by the number of observation times Δt. If

PRINFG = 0	Nothing is printed in this interval
PRINFG > No. of OBS.	Then only "exceptional information" is printed in this interval (e.g., velocity correction information, etc., see Program Definition - Output Block).

PRINFG = Number less than No. of OBS, then those multiples are printed.

For example,

Case	No. of OBS.	OBS. DELTA	PRINFG
1	6	1	1
2	6	1	2
3	6	1	6

In Case 1, the simulation prints at points 1,2,3,4,5,6.

In Case 2, the simulation prints at points 2,4,6 plus "exceptional information" at points in between.

In Case 3, the simulation prints at point 6 only plus "exceptional information" at the other points.

An added feature of this table is built into the first entry. Suppose it is desirable to make an observation at $t = t_o$. It can be accomplished by setting the No. of OBS. = 1, OBS. DELTA = 0, and PRINFG = 1, or 0, depending upon whether one wants a print at $t = t_o$ or does not want a print there. If it is undesirable to make an observation at $t = t_o$, one simply does not input this special first entry.

An important feature of the observation schedule is the input constant specifying the number of entries in this table. The constant works in this manner.

Suppose that one wishes to terminate a run before t_A (time of arrival). This can be accomplished by specifying the number of entries constant so that a subset of the overall observation schedule is used until the time in question is reached. Merely making the observation schedule shorter is not adequate, since without this constant, the last entry in the shorter schedule will be used again and again until t_A is reached. If the observation schedule is sufficiently long to reach t_A and one does not wish to terminate prematurely, it is not necessary to input this constant. However, as a good practice, it is recommended that this constant should be always input.

4.2.2 Stacking of Simulation Runs

To study the effects of a single parameter or a group of parameters, one would normally stack simulation runs. By stacking it is meant that the parameters to be changed are placed behind the "END" card (see input sheets) of the first run and before the "END" card of the second run. The "END" card designates the termination of the input for the run to be simulated. The "END" card has to follow each set of data. The final card of the last case has to be the "FIN" card which follows the "END" card.

4.2.3 Precautions

In stacking runs with the time tables as the trade-off parameters, some caution should be exercised. This is particularly true if the subsequent tables are shorter than the ones they are to replace. What happens is that the shorter table will displace only an equivalent portion of the previous table. If this shorter table is set up so as to cover the entire interval to be simulated, then no problem will occur. However, if this table is to cover a shorter interval, then the remaining interval will use the remaining portion of the previous table.

Special care should be exercised in setting up the observation schedule in stacked runs. This is especially true with respect to the "number of entries" constant. If this constant is always input as recommended, then no problems will occur in stacked runs. However, if one neglects to input this constant every time the schedule is changed, problems could arise. For example, in studying the effects of the observation schedule, one could have a short schedule in the first run and a long schedule in the second run. If one neglects to input the "number of entries" constant in both runs, the number of entries in the first run will govern the second run which will cause the second run to cut off prematurely.

In stacked runs, the noise generator number is different in subsequent runs. Thus, it is almost impossible to reproduce those portions of the program which depend upon the noise generator.

4.2.4 Input Sheets

Program 284

Performance Assessment of Aided-Inertial Guidance and Navigation Systems for Free Fall Page 1 of 12

1 2 3 4 5	7 8 9	11 … 72
9 1	BCI	284, Engineer, Phone, Work Order, Date
9 1	BCI	Ephemeris Tape No., Star Cat. No.
9 1	BCI	Header
9 11	BCI	Header

TRAJECTORY

1 2 3 4 5	7 8 9	11 … 72
1 1	DEC	TL (Time of Launch Double Precision Julian Days) Integral Part, Fractional Part, CT(Time Conversion Factor)
4 2 2 8 0	DEC	TA(Time of Arrival)
3 2	DEC	RVFG, MF(No. of Conics)
4 2 4 0 3	DEC	Planet Radii: RL(0), RL(1), RL(2), RL(3)
4 2 4 0 7	DEC	RL(4), RL(5), RL(6)
3 1 5 9	DEC	Gravitational Consts: MU(0), MU(1), MU(2), MU(3)
3 1 6 3	DEC	MU(4), MU(5), MU(6)

Program 284

Performance Assessment of Aided-Inertial Guidance and Navigation Systems for Free Fall Page 2 of 12

1 2 3 4 5	7 8 9	11 … 72
3 14	DEC	Conic Start Times: T(1), T(2), T(3), T(4)
3 18	DEC	T(5), T(6), T(7), T(8)
3 22	DEC	T(9), T(10)
		Center Flags
3 4	DEC	C(1), C(2), C(3), C(4), C(5), C(6), C(7), C(8), C(9), C(10)
		ROL (Sphere of Influence Radii; NEEDED ONLY WHEN RVFG ≠ 0)
3 85	DEC	ROL(1), ROL(2), ROL(3), ROL(4)
3 89	DEC	ROL(5), ROL(6), ROL(7), ROL(8)
3 93	DEC	ROL(9), ROL(10)
3 95	DEC	Tolerance on ROL: ROT(1), ROT(2), ROT(3), ROT(4)
3 99	DEC	ROT(5), ROT(6), ROT(7), ROT(8)
3 103	DEC	ROT(9), ROT(10)

Program 284

Performance Assessment of Aided-Inertial Guidance and Navigation Systems for Free Fall Page 3 of 12

1 2 3 4 5	7 8 9	11			72
		Initial Nominal Conic States			
		X1*(T1)	X2*(T1)	X3*(T1)	
3 25	DEC	,	,		
		X4*(T1)	X5*(T1)	X6*(T1)	
3 28	DEC	,	,		
		X1*(T2)	X2*(T2)	X3*(T2)	Subsequent States Needed Only When RVFG = 0
3 31	DEC	,	,		
		X4*(T2)	X5*(T2)	X6*(T2)	
3 34	DEC	,	,		
		X1*(T3)	X2*(T3)	X3*(T3)	
3 37	DEC	,	,		
		X4*(T3)	X5*(T3)	X6*(T3)	
3 40	DEC	,	,		
		X1*(T4)	X2*(T4)	X3*(T4)	
3 43	DEC	,	,		
		X4*(T4)	X5*(T4)	X6*(T4)	
3 46	DEC	,	,		
		X1*(T5)	X2*(T5)	X3*(T5)	
3 49	DEC	,	,		
		X4*(T5)	X5*(T5)	X6*(T5)	
3 52	DEC	,	,		
		X1*(T6)	X2*(T6)	X3*(T6)	
3 55	DEC	,	,		
		X4*(T6)	X5*(T6)	X6*(T6)	
3 58	DEC	,	,		

Program 284

Performance Assessment of Aided-Inertial Guidance and Navigation Systems for Free Fall Page 4 of 12

1 2 3 4 5	7 8 9	11 … 72
3 61	DEC	X1*(T7) , X2*(T7) , X3*(T7)
3 64	DEC	X4*(T7) , X5*(T7) , X6*(T7)
3 67	DEC	X1*(T8) , X2*(T8) , X3*(T8)
3 70	DEC	X4*(T8) , X5*(T8) , X6*(T8)
3 73	DEC	X1*(T9) , X2*(T9) , X3*(T9)
3 76	DEC	X4*(T9) , X5*(T9) , X6*(T9)
3 79	DEC	X1* (T10) , X2*(T10) , X3* (T10)
3 82	DEC	X4*(T10) , X5* (T10) , X6*(T10)
2 54	DEC	Actual: PRFG
3 175	DEC	ACTFG
3 168	DEC	Initial State: X1(T1) , X2(T1) , X3(T1) NEEDED WHEN PRFG = 1 and ACTFG = 0
3 171	DEC	X4(T1) , X5(T1) , X6(T1)

1 2 3 4 5	7 8 9	11 — 72
GUIDANCE:		
		TERFG, VELFG
2 27	DEC	
		$\overline{\text{ETA(SQ)}}$, $\overline{\eta^2}$, $\overline{\text{GAM(SQ)}}$, $\overline{\gamma^2}$ rad.^{2}
42267	DEC	
		SIG(GAM), σ^γ, SIG(ETA), σ^η
9 523	DEC	
		R(V_{MIN})
42279	DEC	
		V_{MIN}
4 73	DEC	
		KAP, κ
42274	DEC	1.0
		KAP(1), κ_{11}, KAP(2), κ_{22}, KAP(3), κ_{33}, KAP(4), κ_{44}
42275	DEC	
		X1*(TA), X2*(TA), X3*(TA)
42181	DEC	
		X4*(TA), X5*(TA), X6*(TA)
42184	DEC	
		G1*(TA), G2*(TA), G3*(TA)
42285	DEC	
ELECTROMAGNETIC SENSORS:		
		BSFG
2 2	DEC	
		TRFG
2 31	DEC	

Program 284

1 2 3 4 5	7 8 9	11				72
		HSFG	SSFG			
2 3 5	DEC					
		EPS(S0); ϵ_{SO}				
2 3 4	DEC					
		MINFG	NOSTAR			
2 4 2	DEC					
		W1	W2	W3	W4	
4 7 7	DEC					
		KAP(S), κ_S				
2 4 5	DEC	1.0				
		EPS(W), ϵ_W deg.	EPS(0), ϵ_0 deg.	EPS(1), ϵ_1 deg.	EPS(2), ϵ_2 deg.	
4 2 4 1 0	DEC	180.0				
		EPS(3), ϵ_3 deg.	EPS(4), ϵ_4 deg.	EPS(5), ϵ_5 deg.	EPS(6), ϵ_6 deg.	
4 2 4 1 4	DEC					
		RHO(1MAX)	RHO(2MAX)			
2 4 8	DEC					
		BETA(MAX) deg.	BETA(MIN) deg.			
2 5 5	DEC	90.0	0			
		OME, ω deg./time				
4 2 3 6 0	DEC					
		R(1T), r_{1T}	R(2T), r_{2T}	R(3T), r_{3T}		
4 2 3 6 4	DEC					
		TH(1), θ_1 deg.	TH(2), θ_2 deg.	TH(3), θ_3 deg.		
4 2 3 5 7	DEC					
		PH(1), φ_1 deg.	PH(2), φ_2 deg.	PH(3), φ_3 deg.		
4 2 3 6 1	DEC					
		PS1(0), ψ_{10} deg.	PS2(0), ψ_{20} deg.	PS3(0), ψ_{30} deg.		
2 3 7	DEC					

Program 284

Performance Assessment of Aided-Inertial Guidance and Navigation Systems for Free Fall Page 7 of 12

1 2 3 4 5	7 8 9	11			72
		TIME	C1I	C2I	C3I
* 5 155	DEC	,	,	,	
5 159	DEC	,	,	,	
5 163	DEC	,	,	,	
5 167	DEC	,	,	,	
5 171	DEC	,	,	,	
		R1(0)	R2(0)	R3(0)	
2 7	DEC	,	,		
		SM B(1), b_1	SM B(2), b_2	SM B(3), b_3	
4 3453	DEC	,	,		
		SM B(IO), b_{1o}	SM B(I1), b_{11}	SM B(I2), b_{12}	
4 3444	DEC	,	,		
4 3447	DEC	,	,		
4 3450	DEC	,	,		
		A(I0), a_{1o}	A(I1), a_{11}	A(I2), a_{12}	A(I3), a_{13}
4 3456	DEC	,	,	,	
4 3460	DEC	,	,	,	
4 3464	DEC	,	,	,	
		SPSISQ, $\sigma_{\psi i}^2$ rad.^{2}	SETISQ, $\sigma_{\eta i}^2$ rad.^{2}		
9 24	DEC	,			
9 26	DEC	,			
9 28	DEC	,			

* Maximum number of entries in this table is 50. This table can be extended by adding 4 to the location.

1 2 3 4 5	7 8 9	11		72
		SRIRDI, $\sigma_{\rho i \dot{\rho} i}$	SRIPSI, $\sigma_{\rho i \psi i}$ L.-rad.	SRIETI, $\sigma_{\rho i \eta i}$ L.-rad NEEDED ONLY WHEN R(0) $\neq 0$
9 30	DEC			
9 33	DEC			
9 36	DEC			
		SRDIPI, $\sigma_{\dot{\rho} i \psi i}$ L/T-rad.	SRDIEI, $\sigma_{\dot{\rho} i \eta i}$ L/T-rad.	SPSIEI, $\sigma_{\psi i \eta i}$ rad.^{2} NEEDED ONLY WHEN R(0) $\neq 0$
9 39	DEC			
9 42	DEC			
9 45	DEC			
		R4(0)		
2 10	DEC			

			TIME	R4(11) rad.^{2}	R4(22) rad.^{2}	R4(33) rad.^{2}
*	9 323	DEC				
	9 327	DEC				
	9 331	DEC				
	9 335	DEC				
	9 339	DEC				

			TIME	R4(12) rad.^{2}	R4(13) rad.^{2}	R4(23) rad.^{2} When R4(0) $\neq 0$
*	9 123	DEC				
	9 127	DEC				
	9 131	DEC				
	9 135	DEC				
	9 139	DEC				

* Maximum number of entries in this table is 50. This table can be extended by adding 4 to the location.

Program 284

Performance Assessment of Aided-Inertial Guidance and Navigation Systems for Free Fall Page 9 of 12

1 2 3 4 5	7 8 9	11 TIME	BODY	R(5) rad.^{2}	STAR 72
* 5 705	DEC				
5 709	DEC				
5 713	DEC				
5 717	DEC				
5 721	DEC				
5 725	DEC				
5 729	DEC				

* This table can be extended by adding 4 to the location. (Maximum number of entries in this table is 50)

NAVIGATION:

		SQRTFG				
2 3	DEC					
		MASK				
2 18	DEC					
		K(P)R, K_p^R		K(V)R, K_v^R		
2 20	DEC					
		SIG1, σ_1	SIG2, σ_2	SIG3, σ_3	SIG4, σ_4	SIG5, σ_5
4 101	DEC					
		MOO				
3 174	DEC					
		M11		M22	M33	
9 54	DEC					
		M44		M55	M66	
9 57	DEC					
		WHEN MOO $\neq$ 0: M12		M13	M14	M15 M16
9 60	DEC					

1 2 3 4 5	7 8 9	11 … 72
9 65	DEC	M23, M24, M25, M26
9 69	DEC	M34, M35, M36
9 72	DEC	M45, M46, M56
* 9 525	DEC	TIME, Q11, Q22, Q33, Q44, Q55, Q66
9 532	DEC	
9 539	DEC	
6 4	DEC	B(1)OL, B(2)OL, B(3)OL
9 75	DEC	B_{IL}: B(IL)11, B(IL)22, B(IL)33
9 78	DEC	
9 81	DEC	
9 84	DEC	B(IL)12, B(IL)13, B(IL)23 NEEDED ONLY WHEN B(I)OL $\neq$ 0
9 87	DEC	
9 90	DEC	
6 1	DEC	B(1)0, B(2)0, B(3)0
9 93	DEC	BiI: B(II)11, B(II)22, B(II)33 rad.^{2}, B(II)44 rad.^{2}
9 97	DEC	
9 101	DEC	

* Maximum number of entries in this table is 10. This table can be extended by adding 7 to the location.

1 2 3 4 5	7 8 9	11 ... 72		
		B(II)12	B(II)13 L.-rad.	B(II)14 L.-rad. NEEDED ONLY WHEN B(I)0 ≠ 0
9 105	DEC			
9 108	DEC			
9 111	DEC			
		B(II)23 I/T-rad.	B(II)24 I/T-rad.	B(II)34 rad.^{2} NEEDED ONLY WHEN B(I) 0 ≠ 0
9 114	DEC			
9 117	DEC			
9 120	DEC			
		B(4)0		
6 82	DEC			
		B(4)11 rad.^{2}	B(4)22 rad.^{2}	B(4)33 rad.^{2}
∩ 48	DEC			
		B(4)12 rad.^{2}	B(4)13 rad.^{2}	B(4)23 rad.^{2} NEEDED ONLY WHEN B(4)0 ≠ 0
9 51	DEC			
		B(5) rad.^{2}		
6 92	DEC			

OBSERVATION SCHEDULE

		No. of Entries in Observation Table (MAX 50)		
5 1	DEC			
		No. of OBS.	OBS. DELTA	PRINFG
5 5	DEC			
5 8	DEC			
5 11	DEC			
5 14	DEC			

1 2 3 4 5	7 8 9	11 … 72
		Observation Table (con't)
		No. of OBS. / OBS. DELTA / PRINFG
5 17	DEC	
5 20	DEC	
5 23	DEC	
5 26	DEC	
5 29	DEC	
5 32	DEC	
5 35	DEC	
5 38	DEC	
OUTPUT:		
		PTRJFG / PSTRFG / PGIDFG / PEMSFG / PNAVFG / PLINFG
7 1	DEC	
		CØRDFG
7 11	DEC	
		ØUFG / ØUL / ØUT
9 21	DEC	
		IUFG / IUL / IUT
8 1	DEC	
		EUFG / EUL / EUT
8 4	DEC	
	END	
	FIN	

4.3 TRAJECTORY AND STATE TRANSITION BLOCK SIMULATION

4.3.1 Nominal Block and State Transition Block

In order to simulate the nominal trajectory and the state transition blocks, it is necessary to input only the data and appropriate logic flags pertaining to these blocks. As was stated in the program definition, the logic is set up in such a manner as to automatically bypass the remaining blocks. Thus, most of the necessary input data and flags for simulating these two blocks are delineated on the first four pages and page 12 of the input sheets (see 4.2.4). All of the symbols used in the input sheets are defined in the program definition and will not be repeated here.

4.3.1.1 Sample Input

To get a better feeling of how the input is set up, an example is in order. The example chosen is an Earth-to-Mars leg of a Martian flyby trajectory. The input data as it appears after key punching follows. Reference should be made to the input sheets, (see Section 4.2.4) for the corresponding symbols.

```
9    1 BCI 284,ENGINEER,PHONE, WORK ORDER, DATE
9    1 BCI ,EPHEMERIS TAPE NO. 2 , STAR CAT. NO.
9    1 BCI ,MARTIAN FLYBY, EARTH-SUN-MARS LEG, 1973 PERIOD,NOM.+ST.TRANS.
9   11 BCI ,INPUT/OUTPUT UNITS KM. AND SEC., INTERNAL UNITS KM. AND HRS.

1    1 DEC 2441908.,.5,.11574074E-4
42280 DEC 8121600.1
3    2 DEC 0,3
42403 DEC 6378.1650,          ,621816. 8
42407 DEC 3400.0880
3  159 DEC .39860320E06,,.13271544E12
3  163 DEC .4297780E05
3   14 DEC 0.0,134058.54,8060895.9,8121600.1
3    4 DEC 0,2,4
3   25 DEC 435.95227,-6482.3107,-928.74152
3   28 DEC 12.598543,.47777418,2.5604613
3   31 DEC 12045504.E1,-84587704.,-36691899.
3   34 DEC 22.785127,25.497559,10.941695
3   37 DEC -423758.92,-377794.47,-108552.63
3   40 DEC 6.9124454,6.2542770,1.7937562
5    1 DEC 4
5    5 DEC 1,0.0,1
      1DEC 1,1.0E5,1
      1DEC 2,3.9E6,1
      1DEC 2,2.0E5,1
7    1 DEC 1,1,0,0,0,0
9   21 DEC 1,1.0000,3600.
8    4 DEC 1,1.0000,3600.
8    1 DEC 1,1.0000,3600.
       END

       FIN
```

The first data card containing the Julian Date data (T_L and C_T) is not really necessary in this situation since the RVFG = 0 and the electromagnetic sensor block is not simulated. However, it is included here to show how it is input. Note that T_L = 2441908.5000000 is input in two parts with two decimal points—one before and one after the comma. Since the input time is in seconds and the ephemeris time is in days, the C_T constant equals 1/86400. The time of arrival (T_A) is in seconds which is consistent with the input time. The RVFG = 0 indicates that the initial conditions for each conic are input and not generated in the program. The RVFG = 1 option of this program should not normally be used, since it is more costly and time consuming to generate trajectories here than in Programs 281 and 291 which are specifically designed for this purpose.

The planet radii R_L(I) and gravitational constants MU(I) have to be input in the locations indicated on the input sheets. In this case under consideration, only the Earth (I = 0), Sun (I = 2), and Mars (I = 4) radii and gravitational constants are required; however, it is imperative to either input zero (0) or the actual values for the other constants on the same card so as to preserve the proper locations. Note that if nothing is entered in the other locations, but two commas are key punched in succession, this is equivalent to having a zero between the commas. Next, the conic start times and the central body codes are input. The conic start times are in input time units (seconds in this case). The central body codes in each conic can take on the values, 0, 1,...,6. Since, in this case, the central body of the first conic is the Earth, the second conic the Sun, and the third conic Mars, the center flags are 0, 2, 4, respectively.

The components of the initial position and velocity of each conic are input next. The only remaining inputs are some sort of observation schedule, (i.e., time advance schedule), the output flags, and the time conversion constants. The observation schedule is normally governed by the aiding instruments and its setup is described in more detail in Section 4.2.1.1. In simulating only the trajectory and state transition blocks, the observation schedule is simply governed by time advance and print desirability considerations. In this example, it was chosen to have a print at t_o, 1 at 100,000 seconds, two prints every 3.9E6 seconds and the remaining prints every 200,000 seconds, up to T_A (time of arrival). The output flags are described in more detail in the program definition and will not be repeated here. At this time it is only necessary to say that since only the trajectory and state transition blocks are simulated, the output flags are set so as to print all the information from these blocks. The remaining data cards contain the unit conversion flags and conversion constants. Due to the fact that the input and output units were chosen as km and sec. and the internal units as km and hrs., it is necessary to convert only the time which is accomplished by the constant, 3600 sec/hr.

4.3.2 Actual Block Simulation

The actual trajectory block is simulated by a simple addition to the input, namely, setting the PRFG = 1. In addition to setting the PRFG = 1, it is necessary to establish the source of the initial conditions for the actual trajectory. The ACTFG determines this source, i.e., when ACTFG = 0, the initial conditions have to be input and when ACTFG = 1, the initial conditions are generated using a random Gaussian noise generator and the initial covariance matrix $M(t_o)$. In the later case, the matrix $M(t_o)$ has to be input.

4.3.2.1 Sample Input

The additional data necessary (as it appears after key punching) follows. This data has to be placed before the END and FIN cards above.

```
9   11 BCI  ,ADDITIONAL  INFORMATION NEEDED FOR ACTUAL WITH ACTFG=0
2   54 DEC  1
3  175 DEC  0
3  168 DEC  437.0,-6480.0,-930.0
3  171 DEC  12.596,.475,2.563
```

```
9   11 BCI  ,ADDITIONAL  INFORMATION NEEDED FOR ACTUAL WITH ACTFG=1
2   54 DEC  1

3  175 DEC  1
3  174 DEC  0
9   54 DEC  9.0,9.0,9.0
9   57 DEC  .25E-4,.25E-4,.25E-4
```

Note that in the second case the $M(t_o)$ matrix was chosen to be diagonal and represents a 1σ error of 3 km and 5 m/sec along each position and velocity coordinate, respectively.

4.3.3 Precautions

In simulating two-dimensional planar flights, the nominal and actual trajectories must be input in the X_1 and X_2 plane when the horizon sensor is used as an aiding instrument.

4.4 ELECTROMAGNETIC SENSOR SIMULATION

This program has the capability of simulating the following ground based and on-board electromagnetic sensors as the aiding instruments:

1. Three ground trackers, each capable of measuring range, range rate, elevation, and azimuth angles
2. (a) Horizon sensor, capable of measuring the two local vertical angles and a planet's subtended angle

   (b) Horizon sensor used as a planet tracker, capable of measuring only the two local vertical angles
3. Space sextant, capable of measuring the angle between a star and a planet's limb or a planet's center.

Each aiding instrument can be simulated individually or in combination with another instrument, the particular combination being established by the instrument flags which are input.

The simulation of the electromagnetic sensors without simulating navigation and/or guidance does not yield much information. Thus, one would not normally attempt to simulate this block without the others. For purposes of illustration, however, the inputs below are given only for this block.

4.4.1 How to Simulate Ground Trackers as the Aiding Instruments

The trackers can be simulated by setting the tracker flag (TRFG) to the proper value in the input. The tracker flag can have the following values:

$$\text{TRFG} = \begin{cases} 0, & \text{no ground trackers} \\ 1, & \text{one ground tracker} \\ 2, & \text{two ground trackers} \\ 3, & \text{three ground trackers} \end{cases}$$

In addition to the TRFG, it is necessary to input the appropriate data pertaining to the tracker or trackers. This data consists of the tracker placement (longitude, latitude, radial distance), the accuracy associated with the tracker system (instrument covariance), and the physical constraints (minimum elevation, visibility criteria, etc.).

The trackers are assumed to be located on the Earth, since normally this would be the case. However, one could also simulate tracker locations on any other planet through modifications in the input. The modification consists of placing the planet's physical constants in the Earth input locations and setting $T_L = T_{1950}$ so that the GHA

(Greenwich Hour Angle) is zero. In the latter case, one should restrict oneself to simulating only planet approach phases, otherwise one would introduce ephemeris information which is completely erroneous.

4.4.1.1 The Tracker Instrument Covariance Matrices

To incorporate as much flexibility as possible in the program, the tracker instrument covariance matrices are input as follows. Each tracker has associated with it a "covariance flag" [RI(0)] which establishes a priori whether the measurements to be simulated are correlated or uncorrelated. When RI(0) = 0, I = 1, 2, 3, the measurements are uncorrelated and only the auto-covariances have to be input. The range covariance can be input either as a constant or as a polynomial in range. The range rate covariance can be input as a constant or as a polynomial in range and/or range rate. The covariances for the two angles—elevation and azimuth— are input as constants and in radians squared.

When RI(0) $\neq 0$, the measurements are correlated and the additional cross-covariances are input as constant values. Since the matrices are symmetric, it is necessary to input only the upper triangular elements.

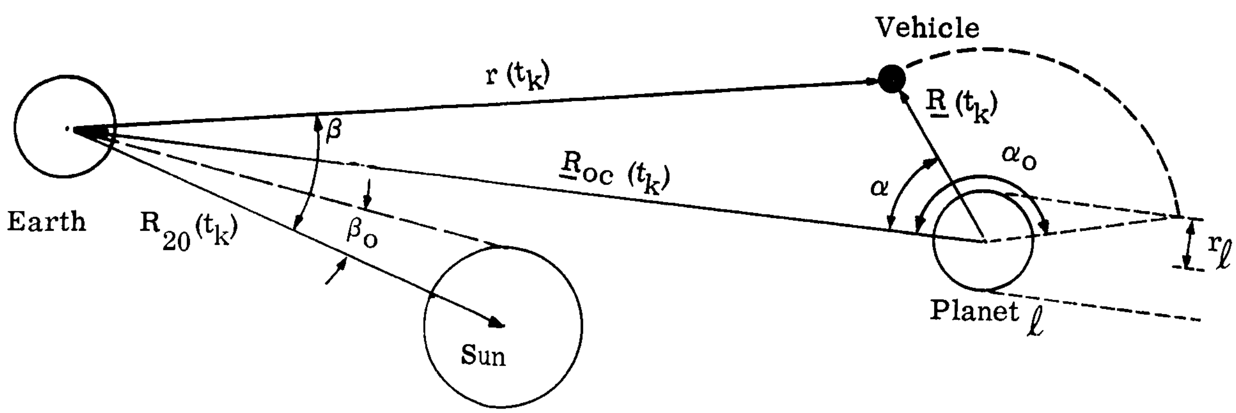
Each one of the tracker covariance matrices is premultiplied by a constant CI which is input in a time table with a maximum of 50 entries. This constant can be used for several purposes. In addition to the elevation angle and visibility criteria, which can automatically eliminate a tracker from the simulation, one could also eliminate any tracker in a prespecified time interval by setting the particular CI constant larger than 1×10^6. A second purpose that these constants serve is in setting up "equivalent" instruments with modified observation schedules. Suppose that one does not wish to use an observation schedule which is in one-to-one correspondence to the correlation times of the trackers. Generally such an observation schedule (e.g., 1 minute sampling interval in DSIF) would be extremely costly to simulate, thus an "equivalent" schedule which preserves the relationship $\sigma^2\Delta t$ = constant can be set up so as to save computer time. The CI constants can then be used to assure that $\sigma^2\Delta t$ remains constant throughout the simulation run, as long as one has less than 50 different values of Δt to contend with.

4.4.1.2 Vehicle Visibility Criteria

The following considerations are taken into account in establishing whether a vehicle is visible from a tracker. When the central body is not the Earth, two simple tests are performed to establish if the Sun is between the vehicle and Earth and if the vehicle is behind the central body planet. The other planets are not taken into account at present in the program. The two tests are: if

$$\epsilon_p \leq \epsilon_{po} \; ; \; \epsilon_s \geq \epsilon_{so}$$

then the vehicle is assumed to be invisible from any tracker on Earth. The figure below defines the above quantities. The behind the planet test is based on the assumption that due to the relative distances involved the shadow of a planet, as seen from earth, is a cylinder rather than a cone.



$$\epsilon_s = \cos\beta \; : \; \epsilon_{so} = \cos\beta_o \; ; \; \epsilon_p = \cos\alpha \; ; \; \epsilon_{po} = \cos\alpha_o = -\sqrt{1 - \left(\frac{r_\ell}{R(t_k)}\right)^2}$$

In addition to the above tests, each tracker is subjected to an elevation angle test to see if the vehicle is above the horizon. Generally, the angle above which a tracker can see the vehicle is taken to be five degrees (i.e., $\psi_{io} = 5^o$). This value is built into the program and can be over-ridden by input also. The maximum elevation angle that is permissible is built into the program as 1.356 radians ≈88 degrees.

4.4.1.3 Precautions

In order to be capable of simulating the range and angular information, it is necessary to input the quantities ρ^1_{max} and ρ^2_{max}. These quantities are not built into the program because they are governed by the unit of length used in the input. Normally these quantities are ρ^1_{max} = 800,000 km and ρ^2_{max} = 400,000 km. Note that when $\rho_i \geq \rho^1_{max}$, $\sigma^2_{\rho i}$ is set to 10^6 (page 3-157) which, in effect, eliminates the range from the measurement through a large variance. The number 10^6 is built into the program and is used irrespective of the internal units. Thus, if the internal units are kilometers the range variance corresponds to 10^3 kilometers (1σ). This is generally satisfactory; however, if it is desired to use a smaller internal unit of length (i.e., meters), the range variance of 10^3 meters would be unsatisfactory. Therefore, the internal constant (10^6) should be modified accordingly. A possible modification is to include this constant in the input unit conversion block.

4.4.1.4 Sample Input

A sample input employing the trajectory given in 4.3.1.1 and using three DSIF trackers is delineated below. Note that the observation schedule is changed from the one given in 4.3.1.1. The purpose of the change is to give a somewhat more "realistic" schedule, as well as to illustrate how one would handle the $\sigma^2 \Delta t$ = constant capability. To illustrate how this run can be terminated before t_A, the "number of entries" constant is input equal to 2 although the observation schedule has 15 entries. With this input, the run will terminate after seven observations (1 observation at t_o and 6 observations on the second entry card).

In this sample, the DSIF accuracies employed are as given in Table 4.1.

RANDOM	BIAS	
σ_ρ = 15 meters	30 meters	Tracker location 100 m. in each coordinate sampling interval 1 minute
$\sigma_{\dot\rho}$ = .03 m/sec.	.05 m/sec.	
σ_ψ = .05 deg.	.1 deg.	
σ_η = .05 deg.	.1 deg.	

Table 4.1 DSIF Error Budget

Note that the tracker accuracies are assumed to be constant and uncorrelated if the sampling interval is larger than 1 minute. The trackers used are located at Goldstone, Johannesburg, and Woomera.

The additional and superseding data required to simulate three ground trackers are:

```
9    1 BCI ,MARTIAN FLYBY, E-S-M LEG, 1973 PERIOD, 3 GR. TRACKER EXAMPLE
9   11 BCI ,ADDITIONAL INFORMATION NEEDED WITH TRFG=3,BSFG=1
2    2 DEC 1
2   31 DEC 3
2   33 DEC -.10,.8999999
2   48 DEC 800000,400000
42360 DEC .41780741E-2
42364 DEC 6378.165,6378.165,6378.165
42357 DEC 243.20505,27.68558,136.88614
42361 DEC 35.06662,-25.73876,-31.21236
2   37 DEC 5.0,5.0,5.0
```

Data continued

```
5  155 DEC 3600.0,.1
      1DEC 10800.0,.33333333E-1
      1DEC 21600.0,.16666667E-1
      1DEC 86400.0,.27777778E-2
      1DEC 172800.0,.13888889E-2
      1DEC 1036800.0,.69444444E-3
      1DEC 7084800.0,.34722222E-3
      1DEC 7948800.0,.69444444E-3
      1DEC 8035200.0,.13888889E-2
      1DEC 8078400.0,.27777778E-2
      1DEC 8100000.0,.55555556E-2
      1DEC 8110800.0,.16666667E-1
      1DEC 8118000.0,.33333333E-1
      1DEC 8125000.0,.1
2    7 DEC 0,0,0
43444 DEC .225E-3
43447 DEC .225E-3
43450 DEC .225E-3
43456 DEC .9E-9
43460 DEC .9E-9
43464 DEC .9E-9
9   24 DEC .76154355E-6,.76154355E-6
9   26 DEC .76154355E-6,.76154355E-6
9   28 DEC .76154355E-6,.76154355E-6
5    1 DEC 2

5    5 DEC 1,0.0,1
      1DEC 1,600.0,1
      1DEC 6,600.0,6
      1DEC 4,1800.0,4
      1DEC 3,3600.0,3
      1DEC 3,21600.0,3
      1DEC 2,43200.0,2
      1DEC 10,86400.0,10
      1DEC 35,172800.0,35
      1DEC 10,86400.0,10
      1DEC 2,43200.0,2
      1DEC 2,21600.0,1
      1DEC 2,10800.0,4
      1DEC 3,3600.0,4
      1DEC 4,1800.0,6
      1DEC 10,600.0,4
```

Since in this example the BSFG $\neq 0$, it is also necessary to input the initial covariance matrices associated with the tracker biases. This additional information is listed under navigation in the input sheets 4.2.4 and is given here.

```
6    4 DEC 0,0,0
9   75 DEC .01,.01,.01
9   78 DEC .01,.01,.01
9   81 DEC .01,.01,.01
6    1 DEC 0,0,0
9   93 DEC .9E-3,.25E-8,.30461742E-5,.30461742E-5
9   97 DEC .9E-3,.25E-8,.30461742E-5,.30461742E-5
9  101 DEC .9E-3,.25E-8,.30461742E-5,.30461742E-5
7    1 DEC 1,1,0,2,3,1
```

4.4.2 How to Simulate the Horizon Sensor as the Aiding Instrument

The horizon sensor can be simulated by setting the "horizon sensor flag" HSFG = 1 and supplying the necessary instrument information. In the simulations, this instrument can be employed in two different modes, either as a horizon sensor or as a planet tracker.

In the horizon sensor mode, the instrument simulates the measurement of the two local vertical angles and the half-subtended angle (β^H). The subtended angle also governs the range of applicability of the horizon sensor through the minimum (β^H_{min}) and maximum (β^H_{max}) subtended angle. In the program, $\beta^H_{min} = 0^o$ and $\beta^H_{max} = 90^o$ are built-in internally and can be over-ridden by input. Normally $\beta^H_{min} = 5^o$ and $\beta^H_{max} = 85^o$, depending upon the instrument under consideration. If the measured subtended angle is inside the β^H_{min}, β^H_{max} range (i.e., $\beta^H_{min} < \beta^H \leq \beta^H_{max}$), the horizon sensor information is used, otherwise the horizon sensor is eliminated.

In the planet tracker mode, the instrument simulates the measurement of only the two local vertical angles. In order to accomplish this capability, it is necessary that $\beta^H_{min} \triangleq 0$ at all times and that the variance of the half-subtended angle be input as a large value (see 4.2). In effect, this eliminates the subtended angle measurement.

4.4.2.1 The Reference Body

In simulating the horizon sensor or planet tracker options, the "reference body" (BODY = 0, 1, 2, . . . , 6) to be employed at a specific observation must be specified through input. The input consists of a time vs. BODY table. This table can have a maximum of 50 entries. This table is also used by the space sextant, thus, care should be exercised in setting up this table.

Generally the BODY would be equal to the central body of the conic section being simulated. This is, however, not always the case. For instance, in the heliocentric phase of a trajectory, one might employ either the departing or approaching planet as the BODY. In such a situation, one will probably use only the planet tracker mode since the half-subtended angle is very small.

4.4.2.2 The Horizon Sensor Covariance Matrix

To complete the input for the horizon sensor, one has to specify the instrument accuracy to be used in the simulation. The instrument covariance matrix elements are input in tabular form as a function of time. As in the ground tracking case, a "covariance flag" R4(0) is used to establish whether or not the measurements to be simulated are correlated. If R4(0) = 0, the measurements are uncorrelated and only the auto-covariances corresponding to the two local vertical angles and the subtended angle have to be input. If R4(0) $\neq$ 0, the measurements are correlated and the additional cross-covariances have to be input also. The two input tables have the form

		TIME	R4(11)	R4(22)	R4(33)
9 323	DEC	,	,	,	
9 327	DEC	,	,	,	
9 331	DEC	,	,	,	

and when R4(0) $\neq$ 0

		TIME	R4(12)	R4(13)	R4(23)
9 123	DEC	,	,	,	
9 127	DEC	,	,	,	
9 131	DEC	,	,	,	

etc.

The tables can be extended to a maximum of 50 entries by adding 4 to the addresses. When the horizon sensor is used in the planet tracker mode, it is necessary to modify the covariance matrix elements as follows:

Set R4(33) = large number (i.e., 1×10^6)

R4(13) = R4(23) $\equiv$ 0

In addition to the $\beta^H_{min} < \beta^H \leq \beta^H_{max}$ test which can eliminate the horizon sensor as the aiding instrument, one could also eliminate the horizon sensor or planet tracker in a specific time interval through input. The instrument is eliminated if and only if R4(11) = R4(22) = R4(33) $\geqq 1 \times 10^6$ in the interval under consideration.

4.4.2.3 Precautions

The following precautions should be taken in setting up the horizon sensor/planet tracker.

a. The instrument covariance matrix elements are input in radians squared.

b. The BODY table has to be input.

When using the horizon sensor/planet in a two-dimensional planar case, the nominal and actual trajectory input must be in the X_1 and X_2 coordinates.

4.4.2.4 Sample Input

A sample input employing the trajectory given in 4.3.1 and using the horizon sensor is given below. The observation schedule employed here is as given in 4.4.1.3. The instrument accuracy employed is .1 degree random error for a 10 minute correlation time and .1 degree bias. The $\sigma^2\Delta t$ capability is used. The instrument is used as a horizon sensor in parts of the departure and approach phases and as a planet tracker in the remainder of the trajectory. The "reference body" is BODY = 0,4 which corresponds to the departing planet Earth = 0 and the approaching planet Mars = 4.

The additional and superseding data are

```
9   1 BCI ,MARTIAN FLYBY, E-S-M LEG, 1973 PERIOD, HORIZON SENSOR EXAMPLE
9  11 BCI ,ADDITIONAL INFORMATION NEEDED WITH HSFG=1,BSFG=1
2  31 DEC 0
2  35 DEC 1,0
2  55 DEC 85.0,0
9 323 DEC 3600.0,.30461800E-5,.304618  E-5,.30461800E-5
9 327 DEC 10800.0,.10153922E-5,.10153922E-5,.10153922E-5
9 331 DEC 21600.0,.50769610E-6,.50769610E-6,1.0E6
9 335 DEC 86400.0,.84616020E-7,.84616 20E-7,1.0E6
9 339 DEC 172800.0,.42308010E-7,.42308010E-7,1.0E6
9 343 DEC 1036800.0,.21154000E-7,.21154000E-7,1.0E6
9 347 DEC 4492800.0,.10577000E-7,.10577000E-7,1.0E6
9 351 DEC 7084800.0,.10577000E-7,.10577000E-7,1.0E6
9 355 DEC 7948800.0,.21154000E-7,.21154000E-7,1.0E6
9 359 DEC 8035200.0,.42308010E-7,.423 8010E-7,1.0E6
9 363 DEC 8078400.0,.84616020E-7,.84616020E-7,1.0E6
9 367 DEC 8100000.0,.16923233E-6,.16923233E-6,1.0E6
9 371 DEC 8110800.0,.50769610E-6,.50769610E-6,1.0E6
9 375 DEC 8118000.0,.10153922E-5,.10153922E-5,.10153922E-5
9 379 DEC 8200000.0,.30461800E-5,.30461800E-5,.30461800E-5
5 705 DEC 3600.0,0
5 709 DEC 10800.0,0
5 713 DEC 21600.0,0
```

```
5 717 DEC 86400.0,0
5 721 DEC 172800.0,0
5 725 DEC 1036800.0,0
5 729 DEC 4492800.0,0
5 733 DEC 7084800.0,4
5 737 DEC 7948800.0,4
5 741 DEC 8035200.0,4
5 745 DEC 8078400.0,4
5 749 DEC 8100000.0,4
5 753 DEC 8110800.0,4
5 757 DEC 8118000.0,4
5 761 DEC 8200000.0,4
```

and the initial covariance associated with the horizon sensor biases is

```
6   82 DEC 0
9   48 DEC .304618E-5,.304618E-5,.304618E-5
```

4.4.3 How to Simulate the Space Sextant as the Aiding Instrument

The space sextant can be simulated by setting the "space sextant flag" SSFG = 1. In addition to the SSFG, it is necessary to input data pertaining to the space sextant. This data consists of the instrument accuracy, the "reference body" BODY, and the method of star selection to be used. In the program there exist three different star options that one could use. These options are:

a. Pre-specified input stars when MINFG = 0

b. "Optimum" ideal stars when MINFG = 1

c. "Optimum" realistic stars when MINFG = 2 or MINFG = 3

In simulating the space sextant, one also has the option of measuring the planet's limb-to-star angle or the planet's center-to-star angle. These options are exercised by setting $\kappa_s = 1$ or 0, respectively.

4.4.3.1 The Space Sextant Covariance

The space sextant random instrument error R_5 is input as a function of time. The input is in radians squared. The location of this input is in the TIME vs. BODY table on Page 9 of the input sheets. This table is also used in the horizon sensor case (see 4.4.2.2). Care should be exercised in setting up this table. This is especially true when the $\sigma^2 \Delta t$ = constant capability is used since then the table is also tied in with the observation schedule.

The space sextant can be eliminated as an aiding instrument in a specific time interval by setting $R_5 \geq 1 \times 10^6$ in that interval.

4.4.3.2 The Reference Body

The "reference body" BODY employed by the space sextant is the same as the one employed by the horizon sensor (see 4.4.2.1). Thus, when both on-board instruments are simulated in the same run, they will use the same BODY. The program does not have the capability of using different reference bodies for the horizon sensor and space sextant.

4.4.3.3 Pre-Specified Input Stars

The program has the capability of using pre-specified stars at pre-established time intervals. The star to be used is established from a TIME vs. STAR table which is located within the TIME vs. BODY table. In order to use this option, one needs to input a star catalog that contains the star numbers (STAR). These numbers do not have to be monotonically increasing (see program definition). The star catalogs used can be either from the library of catalogs or one could set up a new catalog. In this option, no testing is done in the simulation to see if a star is behind a planet or otherwise invisible. It is assumed that the stars called out in the input have been selected properly.

4.4.3.4 "Optimum" Ideal Stars

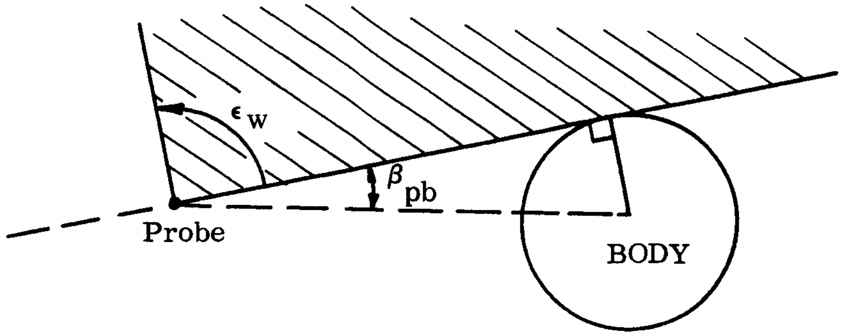
In the "optimum" ideal star option (MINFG = 1), the optimum direction which would minimize the trace of the terminal constraints is established in the simulation. A star is then assumed to exist along this direction. This star is then used in simulating the measurement. Here too, <u>no testing</u> is done to see if such a star (if it existed) could possibly be obscured by other bodies.

The terminal navigation constant matrix used here can be weighted by a diagonal matrix W. With this weighting matrix, one has the capability of weighting the constraints as well as taking into proper account the dimensions involved. The latter is especially important when the constraints have different dimension, e.g., when r, γ, and θ are used. The W matrix is built-in as an identity matrix and can be overridden by input.

4.4.3.5 "Optimum" Realistic Stars

In the MINFG = 2 option, after the "optimal" direction $\underline{h}$ is established, the stars from a realistic catalog are investigated to establish the "best star satisfying a set of constraints (see page 3-171). In the MINFG = 3 option, each star from a realistic catalog is used in computing $Q(\alpha)$, and the one that yields the largest value for $Q(\alpha)$ is used in computing the measurement. Both options select the same stars; however, from the computations involved, MINFG = 3 option is more efficient when the catalog has less than 50 stars.

In addition to the instrument covariance R(5), BODY, and MINFG, one has the capability of specifying tolerances (ϵ = 0,1,...,6) on the subtended angle of the planets. These tolerances are used in the constraint equations. One could also restrict the allowable stars to a specific part of the sky through the input constant ϵ_w. This constant is built-in as ϵ_w = 180 degrees and can be over-ridden by input. For example, ϵ_w = 90 degrees restricts the allowable stars to the forward half of the sky (see Figure below).



4.4.3.6 Sample Input

Sample inputs employing the trajectory given in 4.3.1 and observation schedules given in 4.4.1.3 are outlined below. The instrument accuracy used is .1 degree random error for a 10 minute sampling rate and .1 degree bias. The $\sigma^2\Delta t$ = constant capability is employed here. The reference body used is as given in 4.4.2.4. The additional and superseding data needed in each star option are:

Case a.

```
9    1 BCI  ,MARTIAN FLYBY, E-S-M LEG, 1973 PERIOD, SPACE SEXTANT EXAMPLE
9   11 BCI  ,ADDITIONAL INFORMATION NEEDED WITH SSFG=1,MINFG=0,BSFG=1
2   31 DEC  0
2   35 DEC  0,1
2   42 DEC  0,25
5  705 DEC  3600.0,0,.30461800E-5,5
       1DEC 10800.0,0,.10153922E-5,2
       1DEC 21600.0,0,.50769610E-6,4
       1DEC 86400.0,0,.84616020E-7,12
       1DEC 172800.0,0,.42308010E-7,3
       1DEC 1036800.0,0,.21154000E-7,15
       1DEC 4492800.0,0,.10577000E-7,45
       1DEC 7084800.0,4,.10577000E-7,31
       1DEC 7948800.0,4,.21154000E-7,58
       1DEC 8035200.0,4,.42308010E-7,20
       1DEC 8078400.0,4,.84616020E-7,9
       1DEC 8100000.0,4,.16923233E-6,35
       1DEC 8110800.0,4,.50769610E-6,1
       1DEC 8118000.0,4,.10153922E-5,50
       1DEC 8200000.0,4,.30461800E-5,6
```

```
6    92 DEC .30461766E-5
2    27 DEC 1
42181 DEC -2255.5145,2944.4825,667.50259
42184 DEC 8.5016714,6.1018664,1.8065046
7     1 DEC 1,1,2,2,3,1
```

Star Catalog

STORAGE LOCATION	STAR NUMBER	RIGHT ASCENSION (Radians)	DECLINATION (Radians)	VISUAL MAGNITUDE	SEPARATION MAG.	SEPARATION ANGLE	
42418 DEC	1,	.176028E 01,	-.290796E 00*	-1.58	1.	15.0	With ECL Stars
42421 DEC	2,	.167152E 01,	-.919327E 00*	-.86	1.	15.0	With ECL Stars
⋮			⋮				
43438 DEC	432,	.584528E 01,	-.276537E-01*	3.97	1.0	2.0	WITH ECL STARS
43441 DEC	436,	.504487E 01,	.930187E 00*	3.98	1.0	2.0	WITH ECL STARS

Case b.

```
9     1 BCI ,MARTIAN FLYBY, E-S-M LEG, 1973 PERIOD, SPACE SEXTANT EXAMPLE
9    11 BCI ,ADDITIONAL INFORMATION NEEDED WITH SSFG=1,MINFG=1,BSFG=1
2    42 DEC 1
5   705 DEC 3600.0,0,.30461800E-5
5   709 DEC 10800.0,0,.10153922E-5
5   713 DEC 21600.0,0,.50769610E-6
5   717 DEC 86400.0,0,.84616020E-7
5   721 DEC 172800.0,0,.42308010E-7
5   725 DEC 1036800.0,0,.21154000E-7
5   729 DEC 4492800.0,0,.10577000E-7
5   733 DEC 7084800.0,4,.10577000E-7
5   737 DEC 7948800.0,4,.21154000E-7
5   741 DEC 8035200.0,4,.42308010E-7
5   745 DEC 8078400.0,4,.84616020E-7
5   749 DEC 8100000.0,4,.16923233E-6
5   753 DEC 8110800.0,4,.50769610E-6
5   757 DEC 8118000.0,4,.10153922E-5
5   761 DEC 8200000.0,4,.30461800E-5
42181 DEC -2255.5145,2944.4825,667.50259
42184 DEC 8.5016714,6.1018664,1.8065046
```

Case c.

```
9     1 BCI ,MARTIAN FLYBY, E-S-M LEG, 1973 PERIOD, SPACE SEXTANT EXAMPLE
9    11 BCI ,ADDITIONAL INFORMATION NEEDED WITH SSFG=1,MINFG=2,BSFG=1
2    42 DEC 2,25
42410 DEC 180.0,.1,.05,2.0
42414 DEC .1,.05,.15,.15
```

In Cases a. and c. above, it was assumed that star catalog No. 1 from the catalog library is employed. In Cases b. and c., the terminal constraint conditions must be input, see below.

4.4.3.7 Precautions

When the MINFG = 1 or 2, the trace of the terminal constraints is used to establish an optimum direction. Under these conditions, one must input the TERFG, the nominal state $\underline{X}^*(t_A^*)$, and when the TERFG = 3, the nominal acceleration $\underline{G}^*(t_A^*)$. If the TERFG is set to zero inadvertently the program sets the TERFG = 1 automatically.

4.5 NAVIGATION SIMULATION

In the navigation block simulation, the measurement data generated in the electromagnetic sensor simulation are processed to obtain

a. A new estimate of the state - $\hat{\underline{x}}$

b. The statistics of the error in the estimate - P.

If one simulates more than a single aiding instrument, the data are processed in a sequential fashion, i.e., each instrument is treated independently. In this block, one has the following capability:

1. Process the data via a "regular" Kalman filter
2. Process the data via a "square root" Kalman filter
3. Masking
4. Decide on orbit rectification
5. Introduce a priori statistics error
6. Introduce a "fudge factor" for the computational and dynamic model errors.

4.5.1 Kalman Filter

4.5.1.1 "Regular" Kalman Filter

In simulating the navigation block, one would normally process the data using the "regular" Kalman filter. However, when all the measurements that are to be simulated are uncorrelated, one has also the option of using the "square root" Kalman filter. The "regular" Kalman filter will be used when the SQRTFG = 0. This is always the case unless SQRTFG is input otherwise.

4.5.1.2 "Square Root" Kalman Filter

When the SQRTFG = 1 is input, the program will process the data in the square root mode. In this mode, it is mandatory that all the measurements be uncorrelated, i.e., all the instrument covariance matrices set up in the electromagnetic sensor block are diagonal. In the program there exists no capability of processing the data using both modes in the same simulation run. The advantage of using this mode is that the matrix inversions involved in computing the gain $K(t_k)$ are eliminated. Also this form of computing preserves the symmetry property of the covariance matrix $P(t_k)$. The disadvantage of this option is that it is more costly to simulate when the state has a large dimension.

4.5.2 Masking

The program has the capability of simulating an "equivalent" shorter word length by masking the last number of bits specified by the input MASK. Normally MASK = 0.

4.5.3 Orbit Rectification

The decision regarding the desirability of an orbit rectification is made in the Navigation block. The input constants K_p^R and K_v^R are tested against the normalized position and velocity traces of the covariance matrix to establish if an orbit rectification is desirable. Normally these constants should be input as less than 1 (i.e., $K_p^R < 1$, $K_v^R < 1$) this will assure one that no orbit rectification takes place until the knowledge of the state has improved.

4.5.4 A Priori Statistics Error

In the program one can introduce errors in the a priori statistics through the σ_i, i = 1,2,3,4,5 constants. These σ_i constants are used in computing the noise on the measurements. Note that the σ_i's are tied to the specific aiding instruments, i.e., σ_i is tied to tracker number one, etc. Note also that the same error (σ_i) is used in all the measurements of the same instrument. There is no provision for simulating different a priori statistics errors for the different measurements of the same instrument.

4.5.5 "Fudge Factor" to Account for the Computational and Dynamic Model Errors

To account empirically for the computer round-off and truncation errors and the dynamical model errors, the user of this program has the capability of adding to the variance equation a "fudge factor matrix" Q(t). This matrix is assumed to be a (6 x 6) diagonal matrix. Even when the state vector is augmented, the dimension of this matrix does not change. The elements of this matrix are input in tabular form as a function of time. The input table can have a maximum of ten entries and has the form

		TIME	Q11	Q22	Q33	Q44	Q55	Q66
9 525	DEC	,	,	,	,	,	,	
9 532	DEC	,	,	,	,	,	,	
9 539	DEC	,	,	,	,	,	,	

4.5.6 Precautions

When using the "square root" Kalman filter, one has to be careful with the input. There are no diagnostic tests to establish if the input is consistent with this option. It is up to the individual using the program to make sure that the input is consistent.

When using the "fudge factor" table above, note that there is not enough room on the cards if the Q values have too many significant figures.

4.6 GUIDANCE SIMULATION

The purpose of the guidance block is to simulate the midcourse velocity corrections based on a set of constraints. This block can be simulated by setting the VELFG to the desired value. The VELFG has two functions

a. To set up the proper "terminal constraint flag" TERFG

b. To specify the guidance law to be used in computing the velocity correction.

The VELFG can have six (0, 1,...,5) and the TERFG, four (0,1,2,3) different values. For a detailed definition of the VELFG and TERFG, see the program definition.

It is sometimes desirable to simulate only the terminal constraint statistics without velocity corrections. In such a situation, one inputs the VELFG = 0, as well as the desired TERFG. Note that when the VELFG $\neq 0$, the TERFG is set up automatically and the input TERFG is ignored.

To complete the velocity correction input, one needs the following additional information:

1. The engine characteristics; magnitude $\bar{\eta}^2$ and direction $\bar{\gamma}^2$ variances. One can also input the a priori statistics errors in $\bar{\eta}^2$, σ^{η} and $\bar{\gamma}^2$, σ^{γ}.

2. The nominal state at t_A^*, i.e., $\underline{X}^*(t_A^*)$

3. The nominal acceleration at t_A^*, i.e., $\underline{G}^*(t_A^*)$. This quantity is required only when the VELFG = 4 or 5.

4. The velocity correction criteria κ, $R_{V_{min}}$, κ_i $i = 1,2,3,4$, and V_{min}

Due to the fact that the linear guidance laws employed become singular at t_A^*, the program tests on time ($t_k \geq \kappa t_A^*$) to establish whether the guidance law and velocity correction computations should even be considered. The constant κ is built in as equal to 1. The desirability of performing a velocity correction is based on a statistical consideration. A velocity correction will normally be performed if the quantity $R_V \leq R_{V_{min}}$. Even when the R_V test is passed, the program has the additional option of testing the square root of the diagonal elements of the terminal constraint matrix $\sqrt{T_{E_{ii}}}$. If the $\sqrt{T_{E_{ii}}}$'s satisfy the conditions specified by κ_i again, no velocity correction will be performed. In addition to above tests the magnitude of estimated velocity correction $|\Delta\hat{\underline{V}}|$ is tested against V_{min} and if satisfied the velocity correction is performed.

At present, there is no provision for specifying the velocity corrections at pre-established points along the trajectory.

4.6.1 Precautions

The $\bar{\eta}^2$ and γ^2 are input as the variances of the engine characteristics (not the standard deviations); $\bar{\eta}^2$ is the magnitude variance in $(\text{percent})^2$ and γ^2 is the direction variance in $(\text{radians})^2$.

The $\sqrt{T_{E_{ii}}} \leq \kappa_i$ test is performed in the internal coordinate system when the COORDFG = 1 and the TERFG = 1, thus care should be exercised in inputting the κ_i's.

4.6.2 Sample Input

The additional guidance input, that could be used with any of the preceding inputs, is outlined below. The guidance law employed in this sample corresponds to the fixed position vector and fixed flight time constraints. The engine characteristics correspond to a 1 percent error in velocity magnitude and a .1 degree error in velocity direction. The velocity correction ratio criterion is $R_{V_{min}} = .1$.

```
9     1 BCI ,MARTIAN FLYBY, E-S-M LEG, 1973 PERIOD, SPACE SEXTANT EXAMPLE
9    11 BCI ,ADDITIONAL INFORMATION NEEDED WITH GUIDANCE
2    27 DEC 0,1
42267 DEC 1.0E-4,.304618E-5
42279 DEC .1
42181 DEC -2255.5145,2944.4825,667.50259
42184 DEC 8.5016714,6.1018664,1.8065046
7     1 DEC 1,1,3,2,3,0
```

4.7 OUTPUT

Through the output block, one can control the quantities which one wishes to print, as well as the computations to be performed so as to aid in the interpretation of the results. The desired quantities to be printed in any simulation are specified by a set of output flags. These flags are tied to the functional blocks of the program. For a detailed definition of these flags, one is referred to the program definition.

4.7.1 Output Computations

The computations performed in the output block consist of:

1. Coordinate conversion, when COORDFG = 1. Only a subset of all the quantities is converted to the new coordinates, see the program definition. Care should be exercised in interpreting the results when the state is augmented, since the entire state is not converted.
2. Output Computations
   a. Computation of eigenvalues, eigenvectors, traces, volumes, and vector magnitudes. These computations are governed by the output flags, i.e., if a quantity is not required for print, its corresponding computation is not performed.
   b. Linear approximation, when PLINFG $\neq 0$. This computation is not valid when orbit rectifications and/or velocity corrections are to be simulated.
   c. Total gain computation. The total gain matrix is always computed in the internal coordinates and internal units. Thus, care should be exercised in analyzing this matrix.
3. Output unit conversion, when OUFG $\neq 0$. For output purposes, the program contains the capability of converting the results to a set of units other than the internal ones. The quantities which are converted are governed by the output flags. Under certain circumstances, i.e., PGIDFG = 1, PNAVFG = 1, PEMSFG = 1, no unit conversion in these blocks is undertaken.

4.7.2 Precautions

One should note that not all the quantities are coordinate transformed when COORDFG = 1. <u>One should refrain from using the output flags PGIDFG = 1, PEMSFG = 1, and PNAVFG = 1.</u> These values should be used only when it is necessary to check something out in these blocks, keeping in mind that when these values are used, <u>no unit conversions will take place in these blocks</u>. Care should be exercised in analyzing the total gain matrix since it is always printed in internal program coordinates and units.

4.7.3 Sample Input

```
7    1 DEC 1,1,3,2,3,1
9   21 DEC 1,1.0,3600.0
8    1 DEC 1,1.0,3600.0
8    4 DEC 1,1.0,3600.0
```

4.8 UNSATISFACTORY PROGRAM OPTIONS

At the present writing, the following program options have not been checked out or are inoperative.

1. The Square Root Kalman Filter option used in conjunction with any of the guidance policies is inoperative. To make this option operative, it is necessary to change the subroutine MODESL so that it agrees with the equations on page 3-145 when the SQRTFG $\neq 0$. The Square Root Filter is completely operative in the Navigation mode.

2. The present orbit-rectification option has not proved to operate satisfactorily. To remedy this situation, a major program change would be necessary.

3. Guidance Law #5 has not been completely checked out.

4. The linear approximation of the state is not computed properly after a velocity correction and/or an orbit rectification (present option). To correct this computation, it is necessary to reinitialize the $\underline{x}(t_o)$ and $\Phi(t_k, t_o)$ in Block C.3.2, page 3-73.

5. In the variable time-of-arrival guidance laws, the perturbation at the time-of-arrival $\underline{x}(t_A)$ is computed as the difference between the actual state and the original nominal target state; i.e.,

$$\underline{x}(t_A) = \underline{X}(t_A) - \underline{X}^*(t_A^*)$$

rather than

$$\underline{x}(t_A) = \underline{X}(t_A) - \underline{X}^*(t_A)$$

Because of it, the error in the estimate $\tilde{\underline{x}}(t_A)$ is computed incorrectly <u>only</u> at the time-of-arrival. To correct this computation, it is necessary to use the second version for $\underline{x}(t_A)$ in computing $\tilde{\underline{x}}(t_A)$, i.e.,

$$\tilde{\underline{x}}(t_A) = \hat{\underline{x}}(t_A) - [\underline{X}(t_A) - \underline{X}^*(t_A)]$$

5.0 OPERATOR'S GUIDE: PROGRAM 284

5.1 GENERAL INFORMATION

Program 284 was originally written by the Los Angeles Laboratory of AC Electronics in FORTRAN IV for use on an IBM 7040.

This description and discussion pertains to a particular version of Program 284 that was compiled and executed on the IBM 7094.

It should be noted that any attempt to compile and/or execute the program under any system different from that described below may require modifications.

5.2 MACHINE CONFIGURATION

Program 284 was compiled and executed on an IBM 7094 Mod II computer using an IBM 7090/7094 IBSYS Operating System Version 13.

Table 5-1 contains the unit table configuration used under this system.

Because of its size, Program 284 operates as an overlay job and during execution, the system and the program use almost all of the 32 k storage cells available. Also, during execution four utility tape units (described below) are used in addition to the standard input and output units.

5.3 TAPE USAGE

Program 284 requires four utility units, three of which are referenced by the logical designations 1, 2, and 3. These four units are used as follows:

SYSCK2	The links of the overlay feature are stacked here.
SYSUT1 (logical 1)	An ephemeris tape (if needed) is mounted here.
SYSUT2 (logical 2)	This unit is used briefly by the input portion of the program, but is used primarily as an intermediate output tape on which all possible output is passed in binary form from the calculation portion of the program to the output portion of the program where the desired information is printed.
SYSUT3 (logical 3)	This unit is used to pass the stacked input cases (in both input units and internal working units) from the input portion of the program to the calculation portion of the program.

FUNCTION	SYMBOL	PHYSICAL	LOGICAL FORTRAN IV
Library 1	SYSLB1	A1	
Library 2	SYSLB2	Unassigned	
Library 3	SYSLB3	Unassigned	
Library 4	SYSLB4	Unassigned	
Card Reader	SYSCRD	RDA	
On-line Printer	SYSPRT	PRA	
Card Punch	SYSPCH	A0	
Output	SYSOU1	A3	6
Alternate Output	SYSOU2	A3	
Input	SYSIN1	B3	5
Alternate Input	SYSIN2	B3	
Peripheral Punch	SYSPP1	B4	7
Alt. Peripheral Punch	SYSPP2	B2	
Check Point	SYSCK1	B5	
Alternate Check Point	SYSCK2	B5	
Utility 1	SYSUT1	A4	1
Utility 2	SYSUT2	B1	2
Utility 3	SYSUT3	A2	3
Utility 4	SYSUT4	B2	4
Utility 5	SYSUT5	Unassigned	
Utility 6	SYSUT6	Unassigned	
Utility 7	SYSUT7	Unassigned	
Utility 8	SYSUT8	Unassigned	
Utility 9	SYSUT9	Unassigned	

ATTACHED UNITS NOT ASSIGNED OR RESERVED

A5
A6
A7
A8
A9
B6
B7
B8
B9
B0

INTER SYSTEM RESERVE UNITS

None

Table 5-1. Version 13 Unit Table Configuration

Since units SYSCK2, SYSUT2, and SYSUT3 are written and read by the system and the program during the run, no special considerations need be given them except that SYSUT2 has a full reel of tape mounted on it for large jobs.

Since SYSUT1 contains the prewritten ephemeris tape which will be read via FORTRAN, the density at which the tape is written and the density at which the unit is set should be compatible with the FORTRAN file defined for the unit. Since an ephemeris tape written at 556 bpi was used on a channel set at the 800 HI/556 LOW density setting, a file card was included to define the file associated with unit SYSUT1 (logical 1) as being low density. This is necessary since FORTRAN files are normally set to high density.

An ephemeris tape is not required for all jobs, and the program will halt for the operator to mount the tape only if it is needed. The halt is a FORTRAN PAUSE 1, which appears as a program halt with 1 in the AC. At this point, the operator must mount the required tape and press START to return control to the program.

The ephemeris tapes used by Program 284 are the standard JPL ephemeris tapes covering the following dates:

EPHEMERIS TAPE	FROM	TO
1	243 3280. 5	244 0584. 5
2	243 9500. 5	244 6796. 5
3	244 5708. 5	245 1548. 5

5. 4 DECK ARRANGEMENT

Basically, Program 284 is divided into three major sections which are referred to as the input section, the calculation section and the output section. Since these names are somewhat deceiving, the following summary is included to show the functional blocks contained in the three major sections and to indicate the not so obvious tasks performed by the input and output sections.

5. 4. 1 Input Section

5. 4. 1. 1 Input Block

a. Data arrangement, e. g. , setting up full matrices from input data.

b. Input unit conversion.

5.4.1.2 General Initialization Block

5.4.2 Calculation Section

5.4.2.1 Two-body Nominal Block

5.4.2.2 Explicit State Transition Block

5.4.2.3 Two-body Actual Block

5.4.2.4 Guidance Block

5.4.2.5 Electromagnetic Sensors Block

5.4.2.6 Navigation Block

5.4.3 Output Section (BLOCK)

5.4.3.1 Coordinate Conversion

5.4.3.2 Output Computations; e.g., Eigenvalues and Eigenvectors

5.4.3.3 Output Unit Conversion

The deck arrangement parallels the division of the program into three major sections insofar as there are three major links which are referred to as the input link, the calculate link, and the output link. There are, however, several other smaller links which are necessary to optimize the loading of the major links, thus reducing execution time.

The program deck is divided into the following seven links:

MAIN LINK (0)	This link contains the master control program that calls the other links, and some utility programs that remain in core for use by the three major links.
UTILITY LINK (1)	This link contains several routines used by the INPUT link and the CALCULATE link, but not by the OUTPUT link.
INPUT LINK (2)	This is the INPUT link that corresponds to the input section.

CALCULATE LINK (3)	This is the CALCULATE link that corresponds to the calculate section, except for the desired version of the Kalman filter needed by the NAVIGATION block and loaded as a separate link.
KALMAN LINK (4)	This link contains the standard Kalman filter used if SQRTFG=0.
SQ. ROOT KALMAN LINK (5)	This link contains the square root version of the Kalman filter used if SQRTFG≠0.
OUTPUT LINK (6)	This is the OUTPUT link that corresponds to the output section.

The deck arrangement for a compilation and execution is as follows: (where the $IBFTC cards represent the decks within the various links).

```
          JUL  22  1966                                          PAGE 001

$IBSYS
$RESTORE
$JOB             FORTRAN IV IBSYS VERSION 13
$ID              A C ELECTRONICS
$*     IF A FORTRAN PAUSE 1 OCCURS (APPEARS AS A PROGRAM HALT WITH A 1 IN
$*     THE AC) MOUNT (EPHEMERIS) TAPE  2  ON UNIT A4
$EXECUTE         IBJOB
$IBJOB           MAP,FIOCS
$FILE            'UNIT01',UT1,UT1,LOW,BIN,NOLIST,BLK=256,READY,INOUT
$IBFTC 284       M94
$IBFTC CHANGD    M94
$IBFTC DATA      M94
$IBFTC CROSSD    M94
$IBFTC PRODD     M94                MAIN LINK
$IBFTC UNITD     M94
$IBFTC PMAG      M94
$IBFTC MLTMTD    M94
$IBFTC RADVCD    M94
$ORIGIN          LOC1,CK2
$IBFTC RANDOM    M94
$IBFTC TRANGD    M94
$IBFTC F411Z     M94                UTILITY LINK
$IBFTC SPINVW    M94
$ORIGIN          LOC2,CK2
$IBFTC INPUTD    M94
$IBFTC DLONG     M94
$IBFTC IMOVED    M94                INPUT LINK
$IBFTC DINPUT    M94
$IBFTC FIXD      M94
```

```
$IBFTC LTERCK   M94
$IBFTC INITZD   M94
$IBFTC INUNTD   M94
$ORIGIN         LOC2,CK2
$IBFTC DOUTPT   M94
$IBFTC RPHIRD   M94
$IBFTC PHID     M94
$IBFTC RVIND    M94
$IBFTC NOMNL    M94
$IBFTC STRANK   M94
$IBFTC STRAND   M94            CALCULATE LINK
$IBFTC ACTUAL   M94
$IBFTC CONICD   M94
$IBFTC SINTAL   M94
$IBFTC MTRAD    M94
$IBFTC MINVD    M94
$IBFTC KEPLRD   M94
$IBFTC MMPYID   M94
$IBFTC ROMATD   M94
$IBFTC WEPHEA   M94
$IBFTC SETRVD   M94
$IBFTC MSKAD    M94
$IBFTC EPHEMD   M94
$IBFTC EVRCFD   M94
$IBFTC ELECTD   M94
$IBFTC USTVGD   M94
$IBFTC GRNTRD   M94
$IBFTC HORIZD   M94
$IBFTC SPSEXD   M94
$IBFTC OPTSRD   M94
$IBFTC ANGLED   M94
$IBFTC TRAKND   M94
$IBFTC OBSERD   M94
$IBFTC QFD      M94
$IBFTC DQFD     M94
$IBFTC GIDNSD   M94
$IBFTC TERCND   M94
$IBFTC VECOSD   M94
$IBFTC MODESD   M94
$IBFTC VELCRD   M94
$IBFTC EXTESD   M94
$IBFTC GIDLAD   M94
$IBFTC 1CALC    M94
$IBFTC 1NAVA    M94
$ORIGIN         LOC3,CK2
$IBFTC SQKALD   M94            SQUARE ROOT KALMAN FILTER LINK
$ORIGIN         LOC3,CK2
$IBFTC KALMAD   M94            KALMAN FILTER LINK
$ORIGIN         LOC1,CK2
$IBFTC OUTPUT   M94
$IBFTC EIGEND   M94
```

```
$IBFTC PRTMTD  M94
$IBFTC UNISBD  M94
$IBFTC PRINTD  M94
$IBFTC ZTINTD  M94
$IBFTC OTSUBD  M94
$IBFTC EIGVED  M94
$IBFTC TOGAND  M94
$IBFTC EXAUPD  M94                 OUTPUT link
$IBFTC ZETALD  M94
$IBFTC CORCND  M94
$IBFTC OTKAKD  M94
$IBFTC HEADLD  M94
$IBFTC BTYSTD  M94
$IBFTC GIDPTD  M94
$IBFTC LNAPTD  M94
$IBFTC NAVPTD  M94
$IBFTC UNICND  M94
$IBFTC EMSPTD  M94
$IBFTC MATCAD  M94
$IBFTC PINPUT  M94
$ENTRY         284                 EOF mark (7-8 in column 1)
'
       DATA CARDS
'                                  EOF mark
$IBSYS
$ENDFILE        SYSPP1
$ENDFILE        SYSOU1
```

While it is necessary that the links maintain the order indicated, the decks within a link may be reordered.

Once a binary deck is obtained from the compilation, each source deck may be replaced with its corresponding binary deck for succeeding runs. If no source decks are included in the deck, some loading time can be saved by using the NOSOURCE option in the $IBJOB card.

5.5 ERROR EXITS

There are three types of errors that could occur during a run, each of which is discussed below.

5.5.1 Input Card Errors

These are errors on the input cards that prevent the input routine from interpreting the data on the card properly, and are usually the result of keypunching mistakes or improperly prepared data.

Upon encountering an error of this type, the input routine prints the card in error followed by the comment "ERROR ON THE PRECEDING CARD."

After the output from any preceding cases has been printed, the job will be terminated with the comments

ERROR IN INPUT
TERMINATE THIS JOB.

5.5.2 Errors Occurring During Calculation

These errors (except for IERRF=2) are caused by either:

1. The nonconvergence of an iterative procedure, or;
2. An attempt to invert a singular matrix.

When an error of this type occurs, all of the output requested for the current case is printed and flow proceeds to the next (if any) case following the comments.

AN ERROR OCCURRED WHILE PROCESSING THE DATA AT TIME XX

THE ERROR CODE IS YY

PROCEED TO THE NEXT CASE.

Note that some or all of the data printed for time XX may be simply the values remaining from the previous time point.

The possible error codes that could be printed in the above format are listed below.

ERROR CODE	EXPLANATION
1	Solution of Kepler's equation did not converge.
2	Time out of range for ephemeris tape.
4	Triangularization error for a covariance matrix of noise for one of the instruments.
5	$H P H^T + R$ could not be inverted in the Kalman filter.
7	Triangularization error in initialization for $M(t_o)$.
8	Triangularization error in initialization for B_{1I}
9	Triangularization error in initialization for B_{2I}.

ERROR CODE	EXPLANATION
10	Triangularization error in initialization for B_{3I}.
11	Triangularization error in initialization for B_4.
14	Triangularization error in initialization for B_{12}.
15	Triangularization error in initialization for B_{22}.
16	Triangularization error in initialization for B_{32}.
17	RVFG≠0 option did not converge.
19	Kepler equation did not converge for actual at t_k.
20	Kepler equation did not converge for nominal at t_k
21	Kepler equation did not converge for nominal at t_f^1
22	Kepler equation did not converge for nominal at t_f^2
23	Kepler equation did not converge for nominal at t_f^3
24	Kepler equation did not converge for nominal at t_f^4
25	Kepler equation did not converge for nominal at t_f^5
26	Kepler equation did not converge for nominal at t_f^6
27	Kepler equation did not converge for nominal at t_f^7
28	Kepler equation did not converge for nominal at t_f^8
29	Kepler equation did not converge for nominal at t_f^9
30	Kepler equation did not converge for nominal at t_f^{10}
31	Φ_2 was singular in Guidance law computation

5.5.3 Abnormal Errors

Examples of error falling in this class are:

1. Overflow or too many underflows (if the system has a limiting number);
2. Time estimate is exceeded and the job is terminated by the system before completion;
3. Tape errors.

Type (1) errors are often the result of bad input quantities, and a study of the input and any outputs obtained will sometimes reveal the cause of the error.

When an "abnormal" error occurs, none or only part of the output placed on the intermediate output tape (SYSUT2) may be printed when the job terminates.

If the output tape (SYSUT2) has been saved, it is possible to print what is on the output tape by remounting the tape on SYSUT2 and executing the program 284P supplied.

6.0 PROGRAMMER'S GUIDE: PROGRAM 284

6.1 INTRODUCTION

This paragraph contains supplementary information to that contained in paragraphs 3.0 and 5.0, COMPUTER PROGRAM DESCRIPTION and OPERATOR'S GUIDE, and is intended for a programmer who has the task of maintaining and modifying the program.

Its goal is threefold:

1) To indicate the correspondence between subroutines and sub-blocks defined in paragraph 3.0;

2) To indicate the levels of calls to the various subroutines; and

3) To provide additional programming information for those sections of the program that are not adequately described in paragraph 3.0.

There is no attempt within this paragraph at completeness with regards to flow-charts, since the flowcharts within paragraph 3.0, together with the FORTRAN listing, provide adequate descriptions for the computational parts of the program.

Particular attention, however, is given to those sections of the program that are of a control nature, especially the controling programs for the three major sections of the program, INPUT, CALCULATE, and OUTPUT.

Finally, it should be noted that the subroutine names listed in paragraph 6.1.1 are true subroutine names while those listed in paragraph 5.0 are deck names (different but not so radically that the proper correspondences cannot be made).

Identification	Subroutine Name	Paragraph 3.0 Reference and Page	Paragraph 6.0 Description Page	Paragraph 6.0 Flowchart Page	Program Listing Page
Main Program	284		6-16	6-6	2
Utility Subroutines	CHANGE		6-17		6
	DATA		6-17		7
	CROSS		6-17		8
	PROD		6-22		9
	UNIT		6-25		10
	PRODM		6-22		11
	MLTMTL		6-21		12
	RADVC		6-22		13
	RAND		6-23		15
	TRANG	3.4.5.3.1; p. 3-176	6-24		16
	F411		6-18		18
	SPINV		6-23		21
Input Section Subroutines	INPUTC (Driver)		6-20	6-27	23
	INPUT		6-19		36
	IMOVE		6-19	6-29	30
	FIX		6-18	6-32	41
	INITLZ	3.3.2; p. 3-44 to 3-62	6-19		45
	LONG	3.3.2.2.3; p. 3-50	6-20		29
	INUNIT	3.3.1.2; p. 3-35 to 3-39	6-20		53
	TERCKL	3.3.2.2.5; p. 3-58	6-24		42
Calculate Section Subroutines	CALCC (Driver)	3.2; p. 3-26	6-16	6-35	143
Nominal and Actual Block Routines	NOMNL	3.4.1.1; p. 3-93	6-22		68
	ROMAT	3.4.1.2.2; p. 3-97	6-23		91
	KEPLER	3.4.1.2.2; p. 3-101	6-20		88

6.1.1 Subroutine List with Cross References to Paragraph 3.0, 6.0 and Program Listing

Identification	Subroutine Name	Paragraph 3.0 Reference and Page	Paragraph 6.0 Description Page	Paragraph 6.0 Flowchart Page	Program Listing Page
	RVIN	3.4.1.2.4; p. 3-103	6-23		67
	ACTUAL	3.4.3.1; p. 3-122	6-16		78
	CONICT		6-17	6-39	81
State Transition Block Routines	STRAN	3.4.2.1; p. 3-107	6-24	6-39	72
	INITAL	3.4.2.2.1; p. 3-109	6-19		83
	STRANS	3.4.2.2.2; p. 3-115	6-24		73
	PHI	3.4.2.2.1; p. 3-110	6-22		64
	MINV		6-21		87
	MMPYIN		6-21		90
	RPHIRT		6-23		63
	MTRA		6-21		86
Guidance Block Routines	GIDNSL	3.4.4.1; p. 3-130	6-18	6-39	124
	EXTESL	3.4.4.2.1; p. 3-132	6-18		135
	TERCNL	3.4.4.2.2; p. 3-136	6-24		126
	GIDLAL	3.4.4.2.3; p. 3-137	6-18		138
	VECDSL	3.4.4.2.4; p. 3-141	6-25		128
	VELCRL	3.4.4.2.5; p. 3-143	6-25		133
	MODESL	3.4.4.2.6; p. 3-145	6-21		130
Electromagnetic Sensor Block Routines	ELECTG	3.4.5.1; p. 3-148	6-17	6-40	100
	GRNTRK	3.4.5.2.4; p. 3-156	6-19		105
	TRAKIN	3.4.5.2.3; p. 3-153	6-24		120
	HORIZG	3.4.5.2.5; p. 3-161	6-19		109
	ANGLES	3.4.5.2.5; p. 3-162	6-16		119
	SPSEXG	3.4.5.2.6; p. 3-164	6-23		111
	OBSERV	3.4.5.2.6; p. 3-165	6-22		121
	USTVG	3.4.5.2.6; p. 3-165	6-25		103
	OPSTR	3.4.5.2.6; p. 3-167	6-22		114
	QF	3.4.5.2.6; p. 3-168	6-22		122
	DQF	3.4.5.2.6; p. 3-168	6-17		123

Subroutine List with Cross References to Paragraph 3.0, 6.0 and Program Listing (con't)

Identification	Subroutine Name	Paragraph 3.0 Reference and Page	Paragraph 6.0		Program Listing Page
			Description Page	Flowchart Page	
Navigation Block Routines	NAVA	3.4.6.1; p. 3-173	6-21	6-41	152
	KALMA	3.4.6.2.2; p. 3-179	6-20		161
	SQKALA	3.4.6.2.3; p. 3-182	6-23		156
	MASKA	3.4.6.2.3; p. 3-184	6-21		97
Ephemeris Routines	SETRV		6-23		96
	EPHEM		6-18		98
	EPHEA		6-18		93
	EVRCF		6-18		99
Output Tape Write Routine	TOUTPT		6-24	6-41	60
Output Section Subroutines	OUTPUT (Driver)	3.3.3.1; p. 3-64	6-22	6-44	167
	INPUTL		6-19	6-48	232
	HEADL		6-19	6-71	204
	CORCNL	3.3.3.2.2; p. 3-68	6-17	6-52	196
	OTKAKL	3.3.3.2.3; p. 3-71	6-22	6-53	200
	MATCAL		6-21		230
	EIGEN		6-18		174
	OTSUBL		6-22		186
	TOGANL	3.3.3.2.3; p. 3-73	6-24		188
	UNICNL	3.3.3.2.4; p. 3-74	6-24	6-46	221
	UNISBL		6-25	6-47	179
	PRINTL	3.3.3.2.1; p. 3-64	6-22	6-50	181
	BTYSTL		6-16	6-65	207
	PRTMTL		6-22	6-70	177
	GIDPTL		6-18	6-55	210
	EIGVER		6-18		187
	EXAUPL		6-18	6-67	191
	EMSPTL		6-18	6-61	226

Subroutine List with Cross References to Paragraph 3.0, 6.0 and Program Listing (con't)

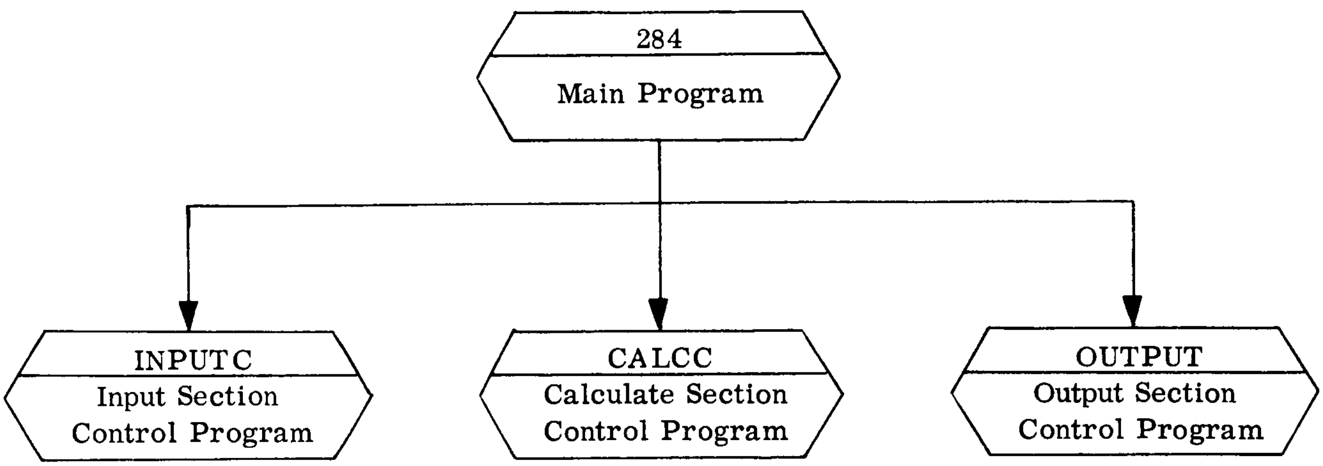
Identification	Subroutine Name	Paragraph 3.0 Reference and Page	Paragraph 6.0 Description Page	Paragraph 6.0 Flowchart Page	Program Listing Page
	ZETAL		6-25		195
	ZTINTL		6-25	6-69	184
	NAVPTL		6-21	6-58	216
	LNAPTL		6-20		214

Subroutine List with Cross References to Paragraph 3.0, 6.0 and Program Listing (con't)

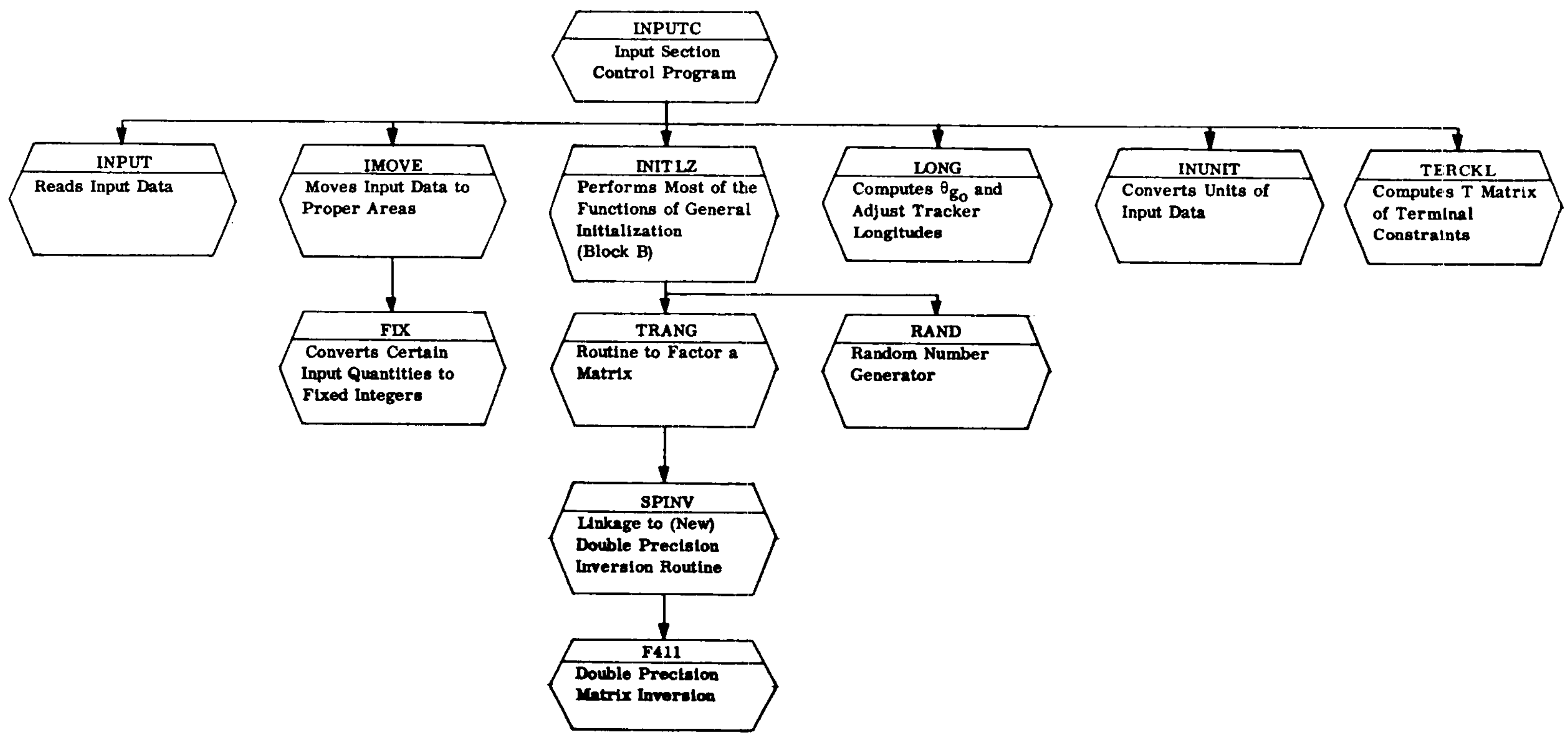
6.2 SUBROUTINE CONTROL CHARTS

It is important to emphasize that this section contains charts showing only what major subroutines are called by what subroutines, and that these charts are not intended to show any logic.

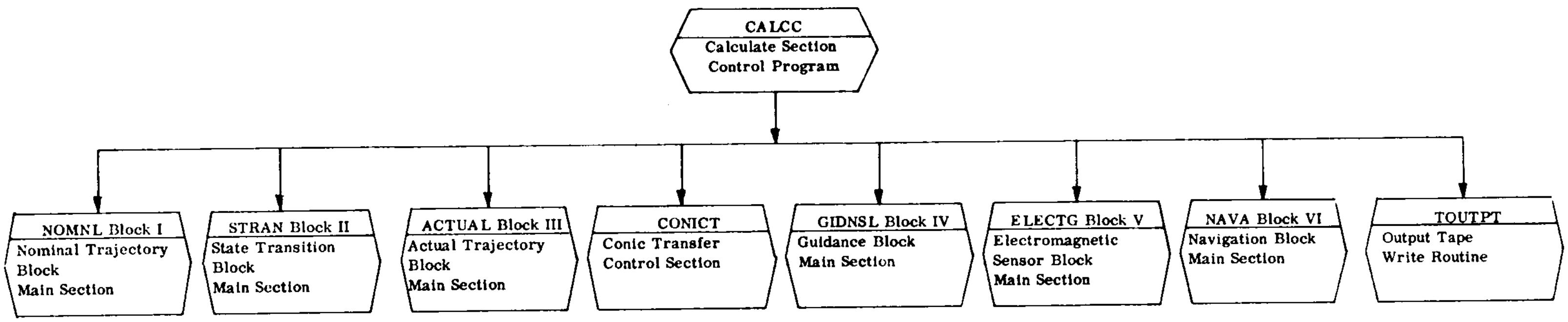
In general, the only order that is indicated is that subroutines on the same level are called in an order proceeding from left to right.



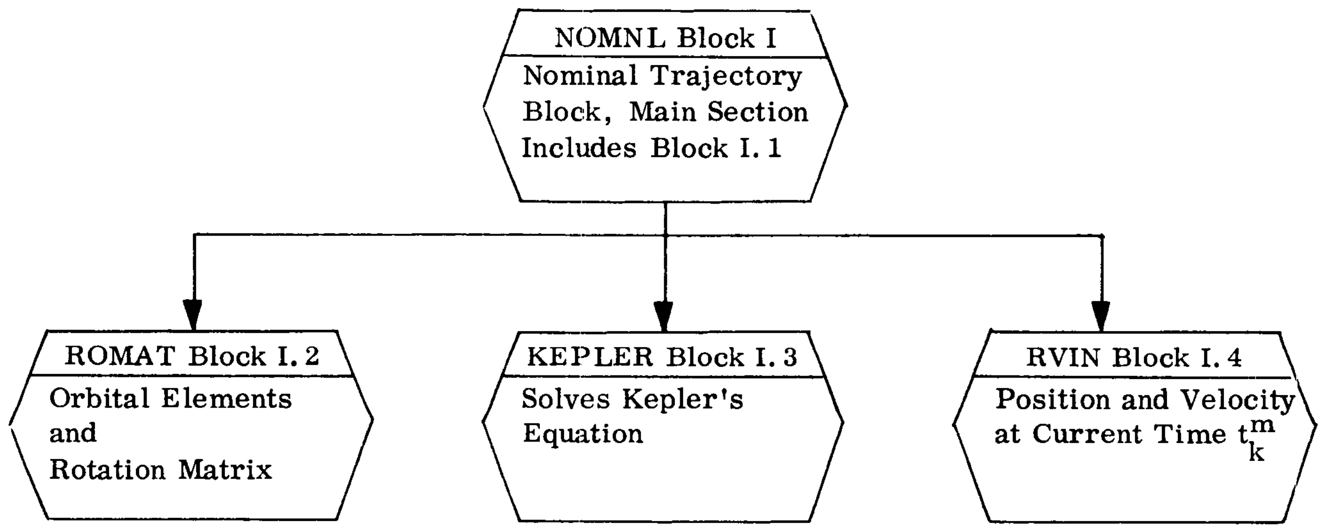
6.2.1 Main Program (284) Subroutine Control Chart



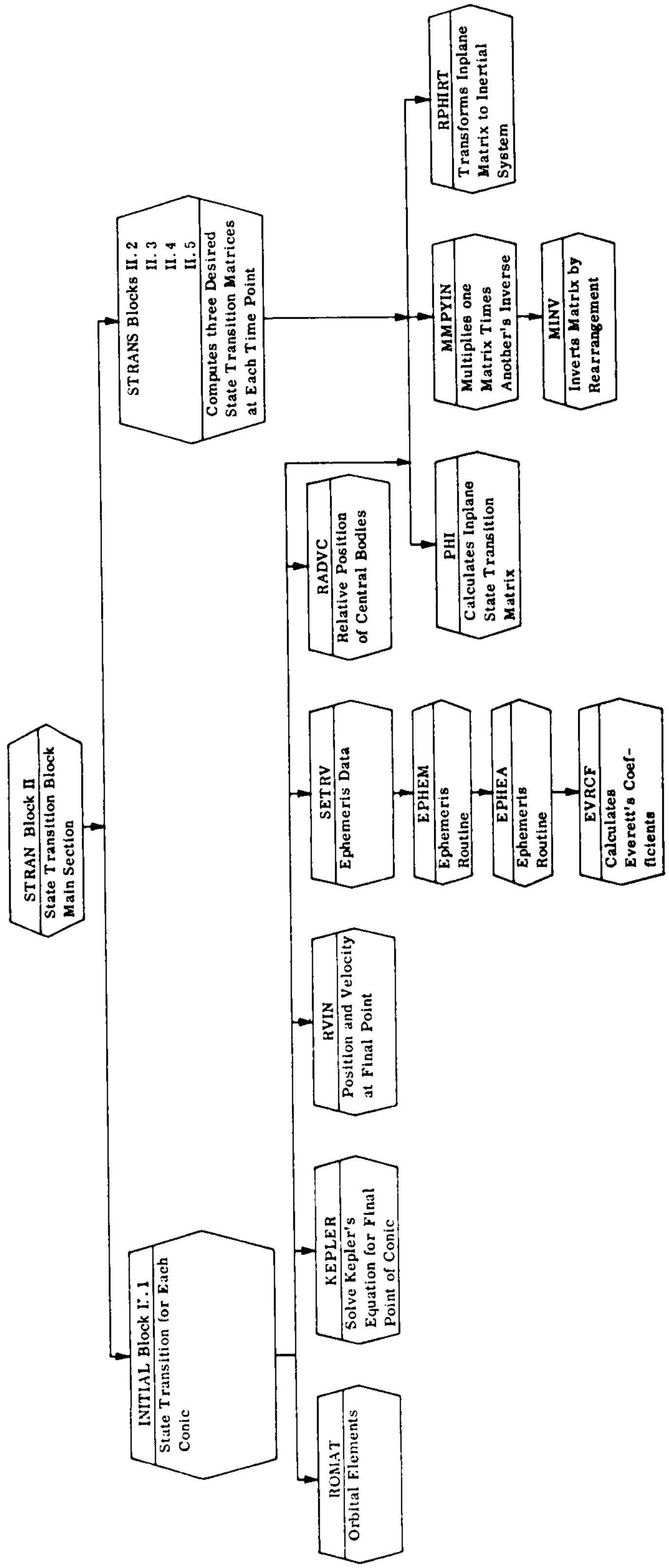
6.2.1.1 Input Section (INPUTC) Subroutine Control Chart



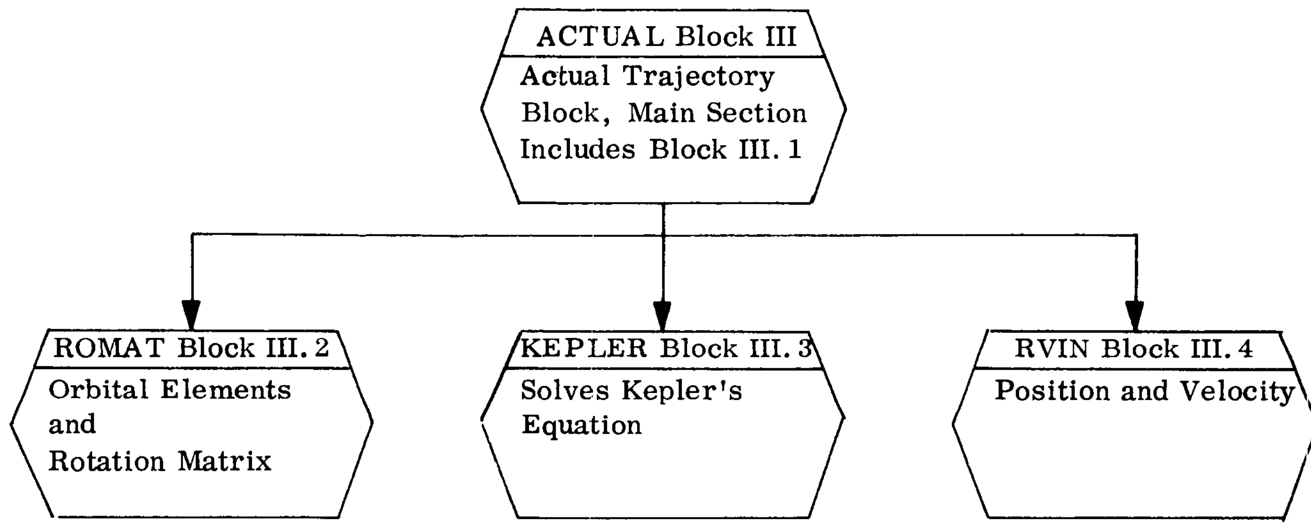
6.2.1.2 Calculate Section (CALCC) Subroutine Control Chart



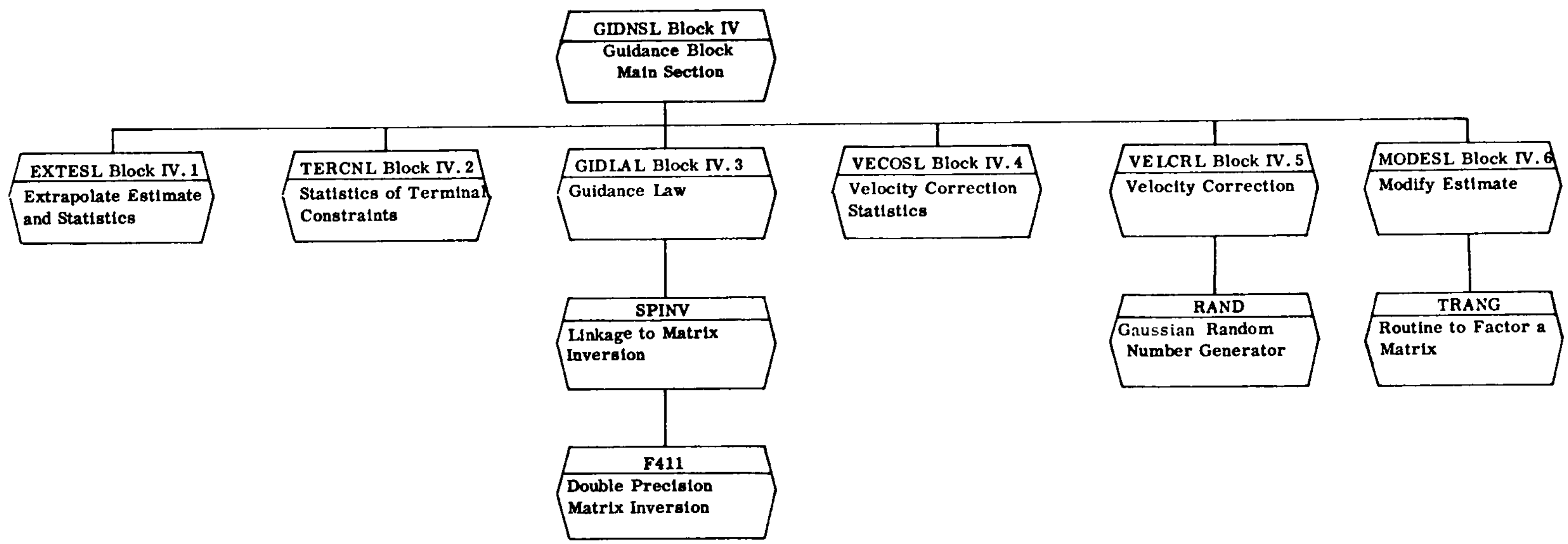
6.2.1.2.1 Nominal Trajectory Block (NOMNL) Subroutine Control Chart



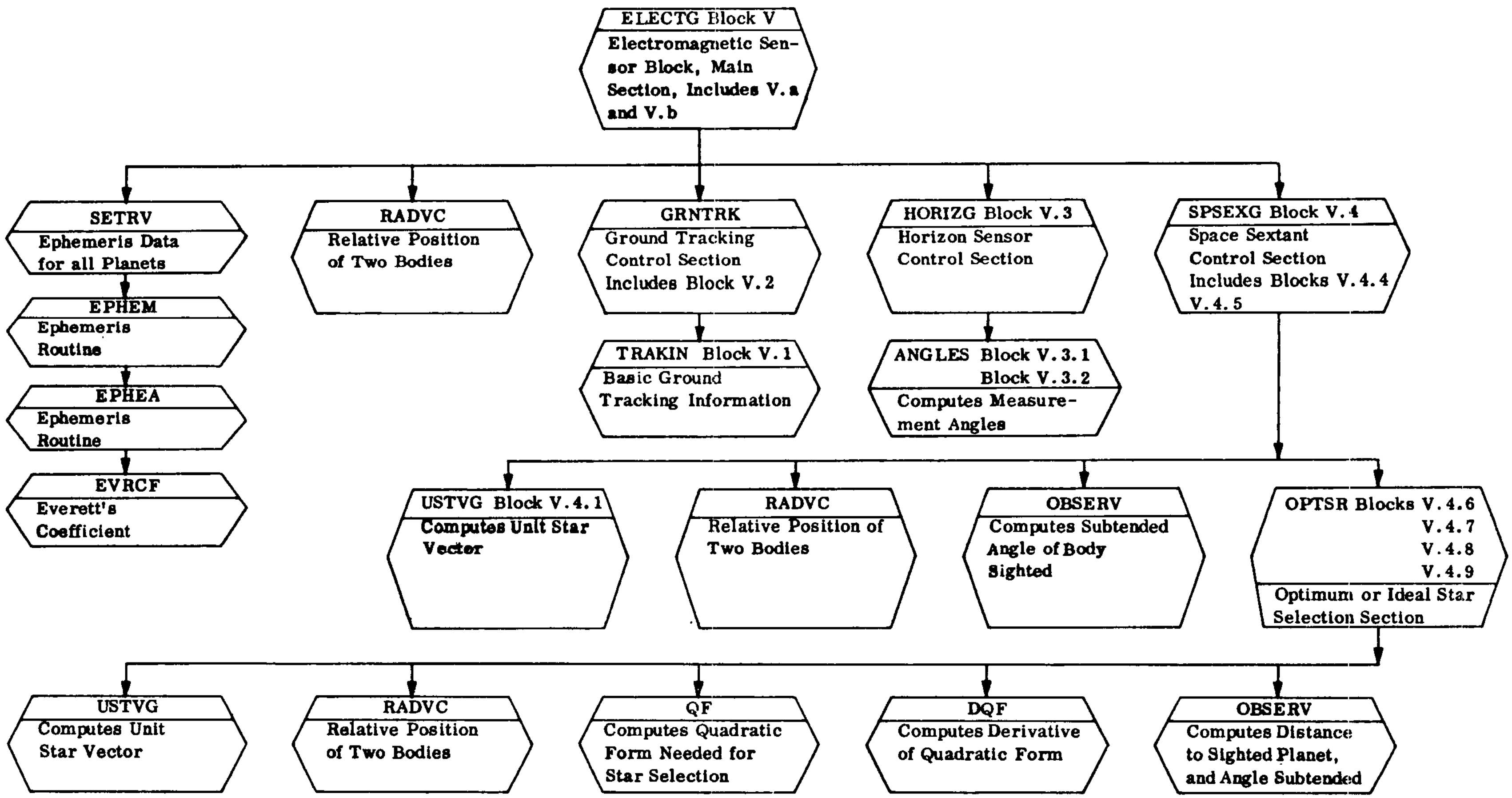
6.2.1.2.2 State Transition Block (STRAN) Subroutine Control Chart



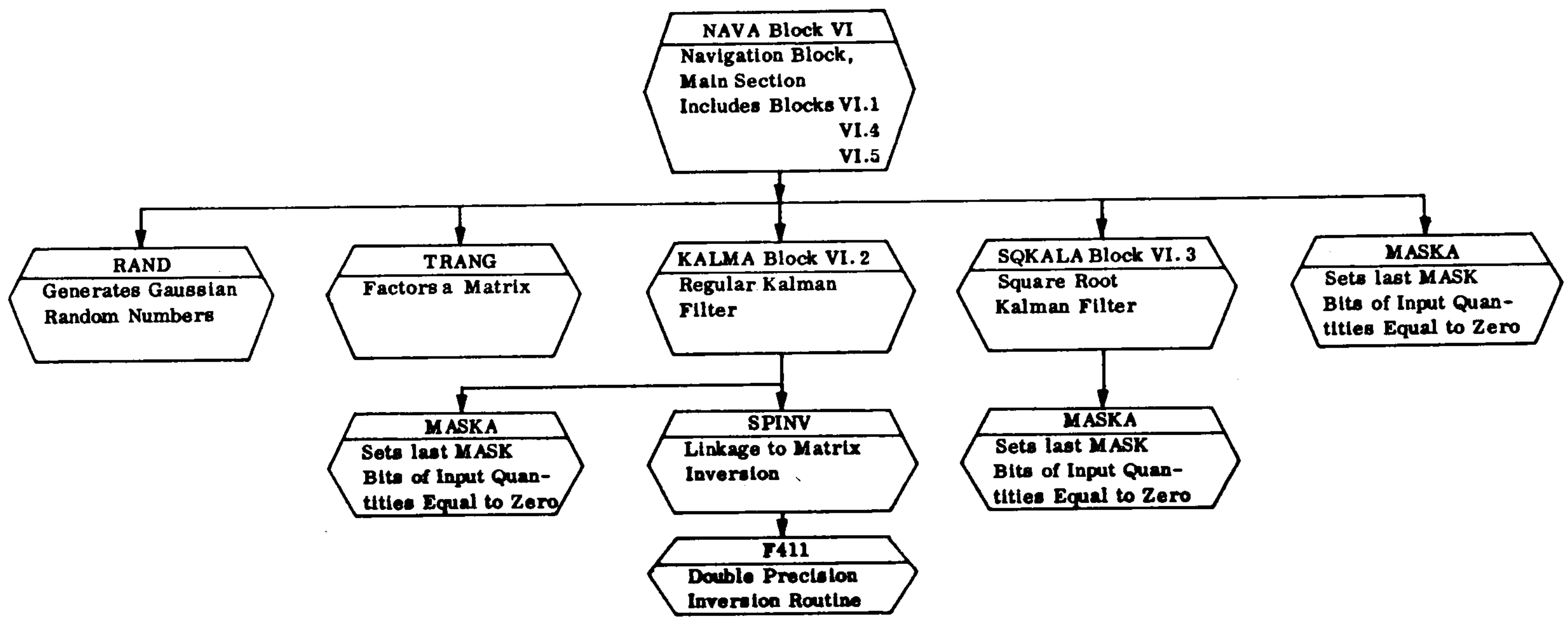
6.2.1.2.3 Actual Trajectory Block (ACTUAL) Subroutine Control Chart



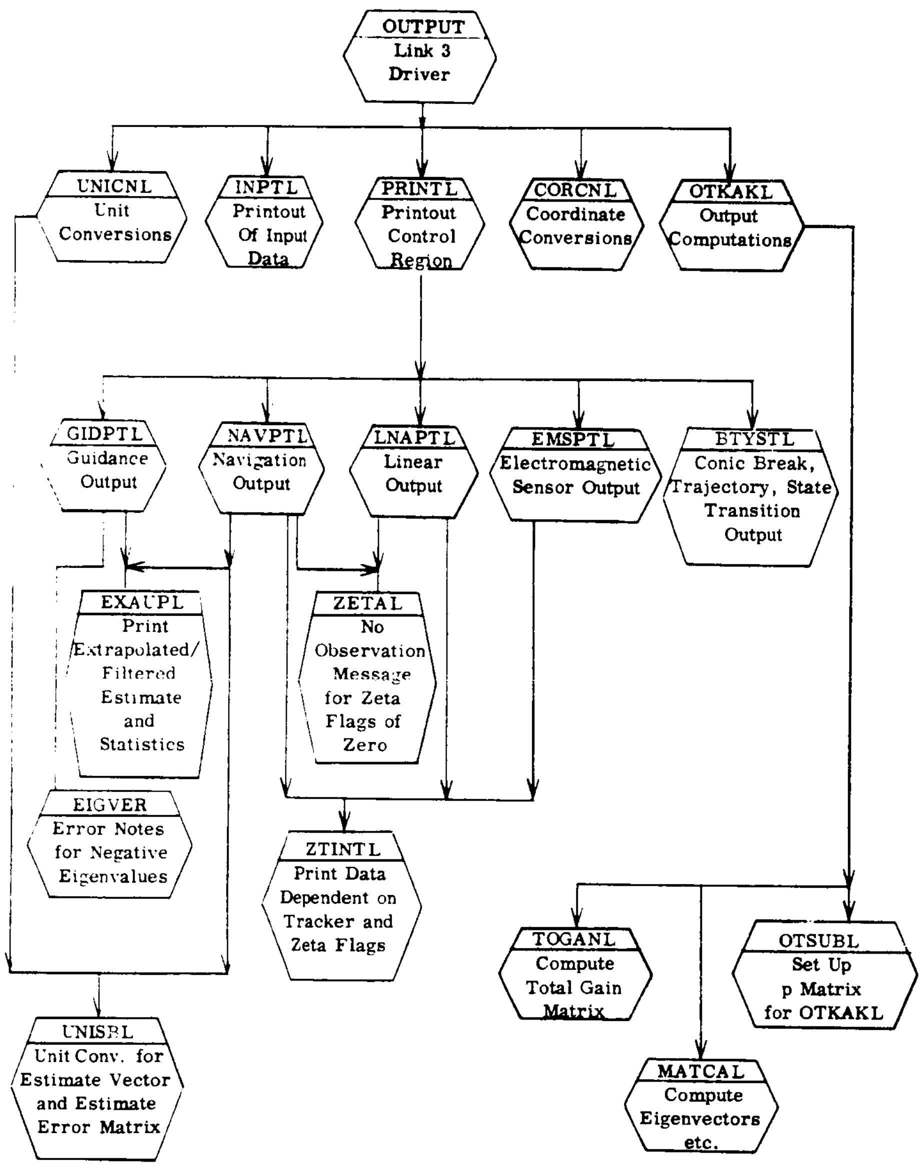
6.2.1.2.4 Guidance Block (GIDNSL) Subroutine Control Chart



6.2.1.2.5 Electromagnetic Sensor Block (ELECTG) Subroutine Control Chart



6.2.1.2.6 Navigation Block (NAVA) Subroutine Control Chart



6.2.1.3 Output Section (OUTPUT) Subroutine Control Chart

6.3 SUBROUTINE DESCRIPTIONS

Routine	Function
284	Main Control Program This program is the master control program that calls the control programs for the three major sections of the program (INPUT, CALCULATE, and OUTPUT).
ACTUAL	Actual Trajectory Block (III) Control Program Controls functions outlined in Block III; computes actual position and velocity at current time, $X(t_k)$.
ANGLES	Horizon Sensor Angles Computations Computes the three angles measured by the horizon sensor.
BTYSTL	Subroutine to print conic break data and/or trajectory data and/or state transition data.
CALCC	Calculate Section Control Program This program controls the functions performed in the calculate section of the program. These functions include: a) reading from the input tape, the input data for each case in both input and internal units, and passing to the output section for output purposes the input in the input units; b) preventing an orbit rectification at the first observation time; c) preventing a velocity correction until an observation has been made; d) performing several initialization functions for each case as required by the trajectory (NOMINAL, STATE TRANSITION and ACTUAL) blocks; e) controlling the time advance f) obtaining from the tables all the quantities needed for each time point g) calling the computational blocks I thru VI for each time point except as follows: i) if there was an error in any block, skip the remaining blocks ii) if there is a velocity correction skip the ELECTRO-MAGNETIC SENSOR and NAVIGATION blocks;

iii) at time of arrival, skip the ELECTROMAGNETIC SENSOR and NAVIGATION blocks;

h) checking for conic transfers;

i) checking for error returns from the computational blocks;

j) passing the output for each time point to the output section via tape, and;

k) returning control to the main program when all cases have been processed.

CHANGE Array Move Subroutine

Simply moves N elements from A to B.

CONICT Conic Transfer Control Program

This routine performs the necessary "bookkeeping" when the vehicle passes from one conic section to another, and aids in the task of introducing an articial time at the conic break as required by the trajectory (NOMINAL, ACTUAL and STATE TRANSITION) blocks.

CORCNL Coordinate conversion subroutines

CROSS Cross Product Subroutine

DATA Block Data Subprogram

DQF Derivative of Quadratic Form

Computes the derivative of the quadratic form used in the optimum star subroutine.

ELECTG Electromagnetic Sensor Block (V) Control Program

This routine controls the functions of the electromagnetic sensor block, which include:

a) determining if ephemeris data is needed at this time point, and if so, obtaining it;

b) determining if the vehicle is visible to the trackers (if used) by applying the visibility test. (Both the ϵ_p and ϵ_s tests are skipped if the central body is the Earth, and the ϵ_s test is skipped if the central body is the Sun.), and;

c) calling the various instrument sections provided the appropriate tests are satisfied.

EMSPTL Subroutine to print electromagnetic sensors

This routine reads data from the JPL ephemeris tapes; it has most of the capabilities of the JPL subroutine EPHEM.

EPHEA Ephemeris Subroutine

This routine reads data from the JPL ephemeris tapes; it has most of the capabilities of the JPL subroutine EPHEM.

EPHEM Ephemeris Linkage Subroutine

This routine is a linkage to the true ephemeris routine EPHEA. It was written to have the same calling sequence as the JPL subroutine EPHEM.

EVRCF Everetts Coefficient Subroutine

Generates coefficients for Everett's equation used for interpolation by the ephemeris routine EPHEA.

EXAUPL Subroutine that is used by GIDPTL to print extrapolated estimate and statistics and that is used by NAVPTL to print filtered estimate and statistics.

EXTESL Extrapolate Estimate and Statistics

This routine performs the task of block IV. 1. It extrapolates from the last point to the current point the statistics matrices M and P and the estimate $\hat{x}$.

F411 Double Precision Inversion Routine

EIGEN Subroutine that computes eigenvalues

EIGVER Subroutine to print errors notes for negative eigenvalues

FIX Fixed Point Conversion Routine

Because all input is read in as floating point numbers, this routine is used to convert certain inputs to their required fixed point form.

GIDLAL Guidance Law Subroutine

Performs the computations of block IV. 3; calculates the Λ matrix and also C_T.

GIDNSL — Guidance Block (IV) Control Program

This routine controls the functions outlined in GUIDANCE block IV. It decides if a velocity correction is to be made and if so calculates the actual velocity correction.

GIDPTL — Subroutine to print guidance output.

GRNTRK — Ground Tracking Control Section

Performs the tasks of blocks V. 1 and V. 2 if ground trackers are used; computes tracker measurements for the nominal and actual trajectories, the tracker observation matrices and the tracker covariance matrices.

HEADL — Subroutine to print heading at the top of each page

HORIZG — Horizon Sensor Control Section

Performs the task of block V. 3. Computes for the horizon sensor:

a) measurements, and;

b) observation matrix.

IMOVE — Input Data Preparation Subroutine

This subroutine performs several necessary functions on the input data, such as:

a) calls subroutine FIX to convert some input quantities to fixed point integers;

b) counts the number of entries in the input tables, and;

c) moves some input from one area to another, including setting up diagonal or symmetric matrices.

INITIAL — State Transition for Each Conic

This routine performs the task outlined in Block II. 1, namely computing the state transition matrix from the beginning to the end of each conic, $\Phi(t_o^{m+1}, t_o^m)$. It is essentially an initialization task needed.

a) at the beginning of each case, and;

b) after an orbit rectification.

INITLZ — General Initialization Subroutine

Performs case initialization for the various blocks as outlined in the GENERAL INITIALIZATION Block B except for sub-blocks B. 3. 1 and B. 5. 2-B. 5. 5 which are performed in LONG and TERCKL, respectively.

INPTUL Subroutine to print input data for each case.

INPUT Card Input Subroutine

INPUTC Input Section Control Program

This program controls the functions performed in the input section of the program. These functions include:

a) zeroing out the input area at the state;

b) setting certain parameters to their standard values;

c) reading the input data for each case;

d) converting certain input quantities to fixed integers;

e) moving all input to its proper place;

f) checking certain input flags for consistency;

g) converting necessary input to internal units;

h) calling the routines to perform general initialization (Block B);

i) passing via tape to the calculate section the input data for each case in both the input units and the internal units;

j) checking for input or initialization errors, and;

k) returning control to the main program when all input has been read in or when an error was detected.

INUNIT Input Units Conversion Subroutine

This routine converts certain angular input quantities from degrees to radians, and performs input units conversion as outlined in Section 3. 3. 1. 2.

KALMA Kalman Filter Subroutine

This routine performs the tasks outlined in Block VI. 2, the Kalman filter. It obtains the best estimate of the state vector based on the latest measurements and quantities extrapolated from the last time point.

KEPLER Kepler Equation Solution

Performs functions of Block I. 3.

LNAPTL Subroutine to print linear approximation

LONG — Earth Tracker Longitudes Subroutine

This routine performs the functions of Section B.3.1, namely;

a) compute Greenwich hour angle at T_L, and;

b) Adjust tracker longitudes to their proper values relative to the inertial reference system.

MASKA — Mask Subroutine

This routine sets last MASK (an input quantity) bits of input quantities equal to zero. It is used to simulate the use of a shorter-word length in the computations.

MATCAL — Subroutine used by QTKAKL to find eigenvector, eigenvalue square roots, volume and trace for a 3 by 3 matrix which is assumed to be nearly symmetric.

MINV — Matrix Inversion by Rearrangement

Obtains the inverse of a state transition matrix by the rearrangement method.

MLTMTL — Matrix Multiplication Subroutine

MMPYIN — Multiply Matrix by the Inverse of Another Matrix

Given two state transition matrices $\Phi(\tau_k, 0)$ and $\Phi(\tau_{k-1}, 0)$, this routine obtains $\Phi(\tau_k, \tau_{k-1}) = \Phi(\tau_k, 0)\ \Phi^{-1}(\tau_{k-1}, 0)$.

MODESL — Modify Estimate Subroutine

Performs computation of block IV.6; if a velocity correction was made, modify M, P, and $\hat{x}$.

MTRA — Matrix Transposition Subroutine

NAVA — Navigation Block (VI) Control Program

This routine controls the functions of the navigation block, which include:

a) calculating measurements (including noise);

b) exercising the desired filter;

c) updating the estimate and P if there was a velocity correction or there were no measurements made at this time point, and;

d) deciding if an orbit rectification is to be made.

NAVPTL — Subroutine to print navigation output.

NOMNL — Nominal Trajectory Block (I) Control Program

Controls functions outlined in Block I; computes nominal position and velocity at current time, $X^*(t_k)$;

OBSERV — Space Sextant Observation Computation

Computes the unit vector from the vehicle to the body being sighted: the distance to the sighted body, and the angle subtended by the body.

OPTSR — Optimum Star Subroutine

This routine computes the direction of an "ideal" star as outlined in blocks V.4.6 and V.4.7, and if the input flag MINFG is greater than 1, finds the "optimum" star from the supplied catalog that satisfies a set of specified constraints.

OTKAKL — Subroutine to perform computation called for during output.

OTSUBL — Subroutine that sets up the 6 by 6 submatrix P(TK) of the covariance of errors in estimate matrix for use by subroutine ϕTKAKL.

OUTPUT — Control program for the output link (#3) of program No. 284.

PHI — Compute Inplane State Transition Matrix $\Phi(\tau_k, 0)$

Performs the computations outlined in Block II.1.2 to obtain $\Phi(\tau_k, 0)$.

PRINTL — Subroutine to control printout.

PROD — Dot Product Function

PRODM — Vector Magnitude Function

PRTMTL — General matrix print subroutine

QF — Quadratic Form Subroutine

Computes a quadratic form used in the optimum star subroutine.

RADVC — Position or Velocity of a Planet

Obtains position or velocity of one planet with respect to another from an array set up by subroutine SETRV containing position and velocity of the planets with respect to the Sun.

RAND Random Number Generator Function

Depending on an input flag K, generates

a) uniformly distributed numbers on (-1, +1), or

b) normally distributed numbers with mean zero and variance one.

ROMAT Orbital Elements and Rotation Matrix

Performs functions of Block I. 2.

RPHIRT State Transition Matrix Transformation Subroutine

Given an inplane state transition matrix $\Phi(\tau_k, \tau_{k-1})$ and the rotation matrix R, this routine obtains the inertial state transition $\Phi(t_k, t_{k-1}) = R\Phi(\tau_k, \tau_{k-1}) R^T$.

RVIN Position and Velocity in Inertial System

Performs functions of Block I. 4.

SETRV Positions and Velocities of all Planets

This routine sets up the position and velocity of all planets and the Moon with respect to the Sun in units of KMS and KMS/SEC or in other units as specified by the Ephemeris unit conversion parameters. The data set up by this routine will then be used by subroutine RADVC.

SPINV Matrix Inversion Linkage Subroutine

Simply provides linkage to a new inversion routine that replaces the old SPINV inversion routine.

SPSEXG Space Sextant Control Section

Controls functions performed by the Space Sextant block V. 4; computes for the space sextant:

a) nominal and actual measurements, and;

b) observation matrix.

SQKALA Square Root Kalman Filter Subroutine

This routine performs the tasks outlined in Block VI. 3, the Square Root Kalman filter.

STRAN State Transition Block (II) Control Program

Controls functions of Block II; computes three state transition matrices at each time t_k, $\Phi(t_k, t_{k-1})$, $\Phi(t_k, t_o)$ and $\Phi(t_A, t_k)$.

STRANS State Transition Matrix Computation Subroutine

This routine performs the major part of the computation leading to three state transition matrices $\Phi(t_k, t_{k-1})$, $\Phi(t_k, t_o)$ and $\Phi(t_A, t_k)$.

TERCKL Terminal Constraints Matrix Computation

Computes terminal constraints matrix T as outlined in Sections B.5.2–B.5.5.

TERCNL Statistics of Terminal Constraints

This routine performs the computations in block IV.2. It computes the T_E matrix and the square root of its diagonals.

TOGANL Subroutine to compute and set up the total gain matrix

TOUTPT Output Tape Write Subroutine

This routine does most of the writing of output quantities onto the output tape. It writes any and all of the quantities that might be output for the current time. In addition to this routine, the following routines write on the output tape:

a) GIDNSL writes the extrapolated P matrix and state vector onto the output tape, and;

b) CALCC writes the extrapolated P and state vector on the output tape if there was an error return from the Nominal, Actual or State Transition blocks simply to maintain a consistent tape format.

TRAKIN Ground Tracking Measurement Subroutine

Calculates the tracker measurements and certain quantities needed to compute the observation matrix and the tracker covariance matrix.

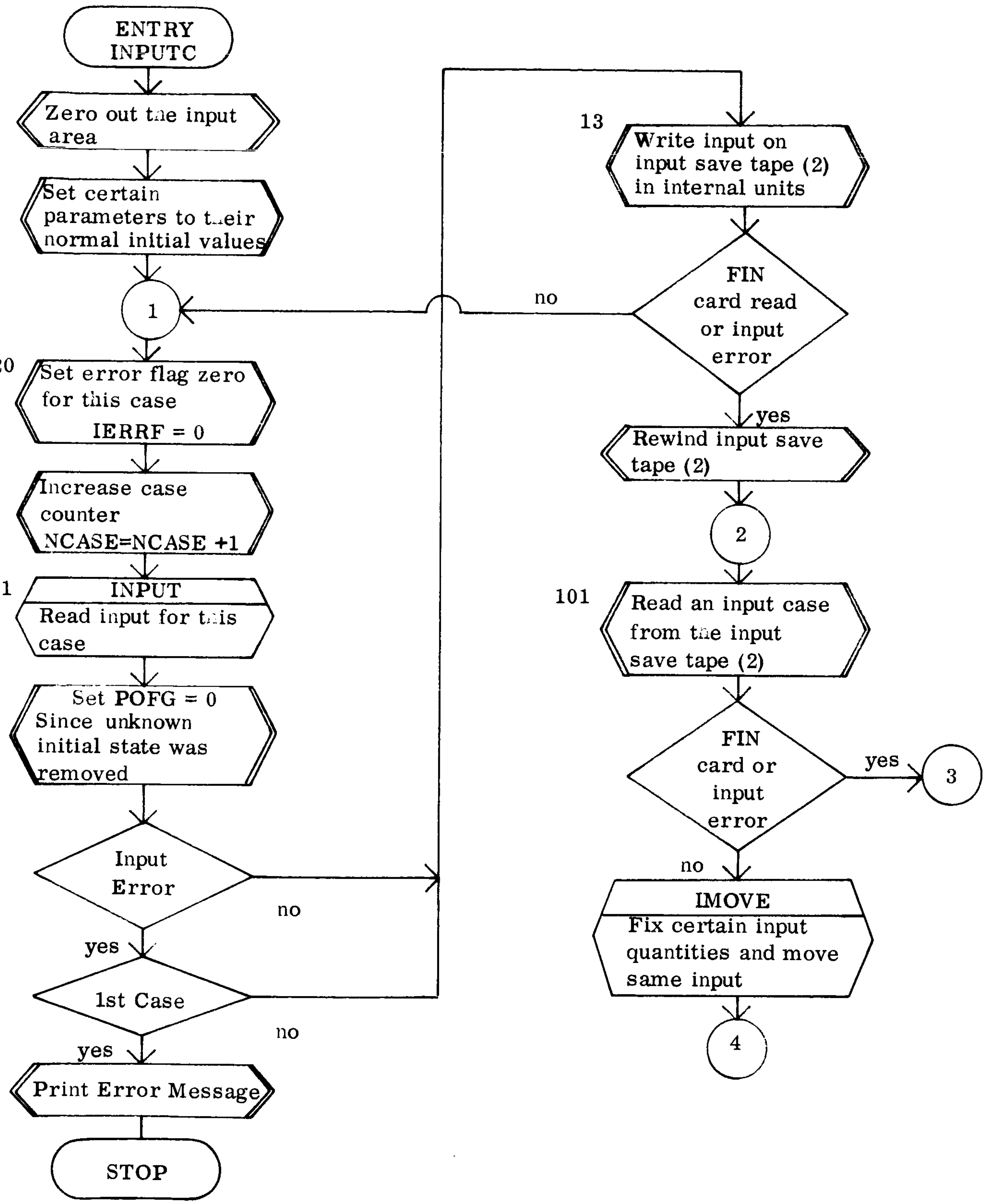
TRANG Matrix Triangularization Subroutine

Given a matrix M, this routine finds matrices T and D such that $M = TDT^T$, where T is lower triangular and D is diagonal.

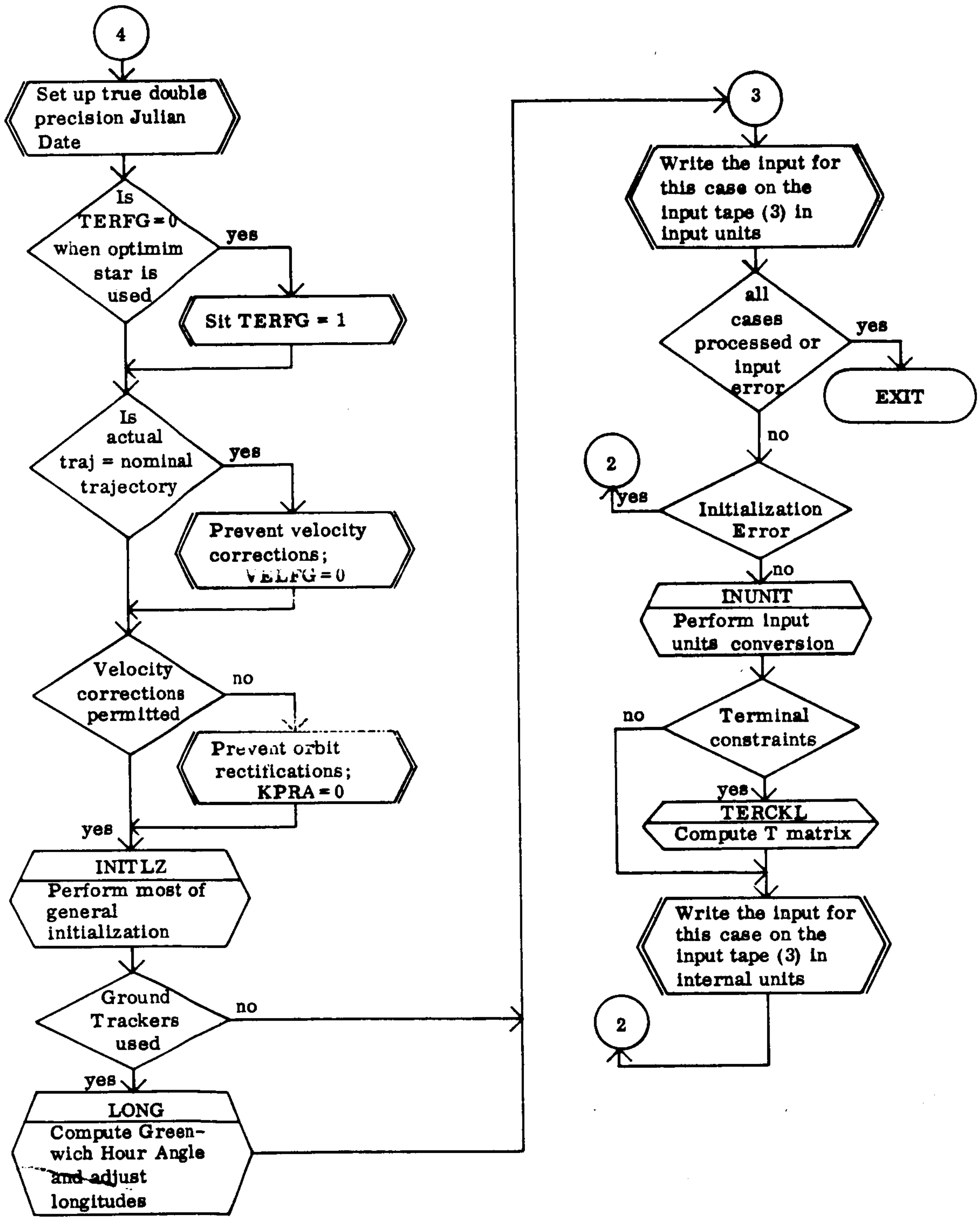
UNICNL Output unit conversion subroutine.

UNISBL	Subroutine that is used by subroutine UNICNL to convert units of best linear estimate vector and covariance of error in estimate matrix.
UNIT	Unit Vector Subroutine
USTVG	Unit Star Vector Computation Given the location of a star in the star catalog, this routine computes a unit vector to the star using the right ascension and declination listed in the catalog.
VECOSL	Velocity Correction Statistics Subroutine Performs the computations of block IV.4; computes statistics that determine if a velocity correction should be made.
VELCRL	Velocity Correction Subroutine Performs the computations of block IV.5; computes the velocity correction if one is to be made.
ZETAL	Subroutine to print the "no observation." message when zeta flags are zero.
ZTINTL	Subroutine that determines the printout of vectors and matrices which depend on the tracker flag and the zeta flags.

6.4 SUBROUTINE FLOW CHARTS



6.4.1 INPUTC - Input Section Control Program



INPUTC - Input Section Control Program (contd)

6.4.2 IMOVE - Input Section Preparation Subroutine

A fairly large number of input quantities are read into a scratch area (array T) and then moved by subroutine IMOVE to their proper storage locations.

This procedure is used for two reasons:

1) to facilitate the desired option to input diagonal and symmetric matrices, and;

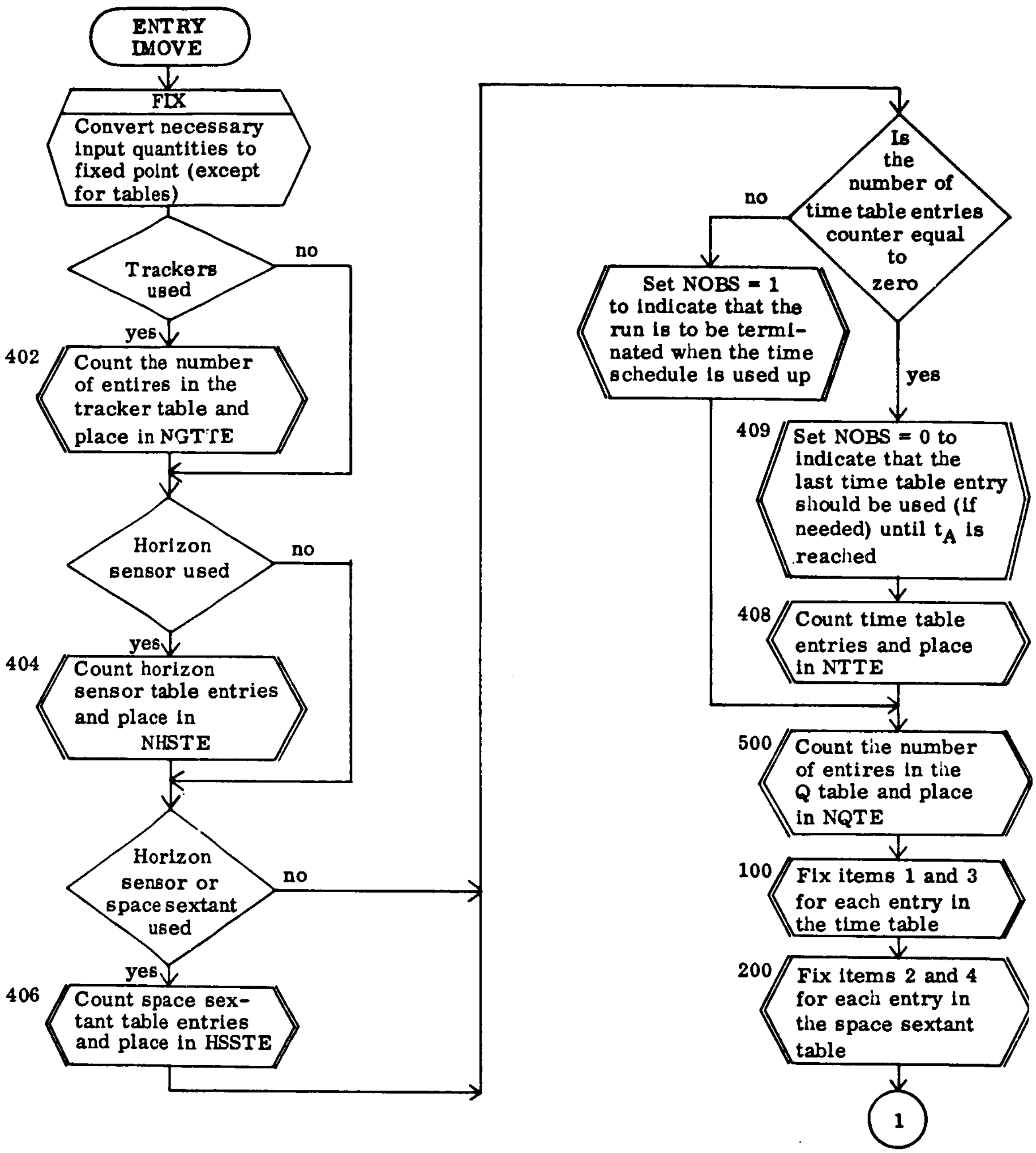
2) to handle additional input quantities that were not provided for in the original storage allocation and the call to the INPUT routine.

In the flow chart, the word "move" is used to refer to quantities corresponding to 2), and the phrase "set up" is used with reference to quantities corresponding to 1).

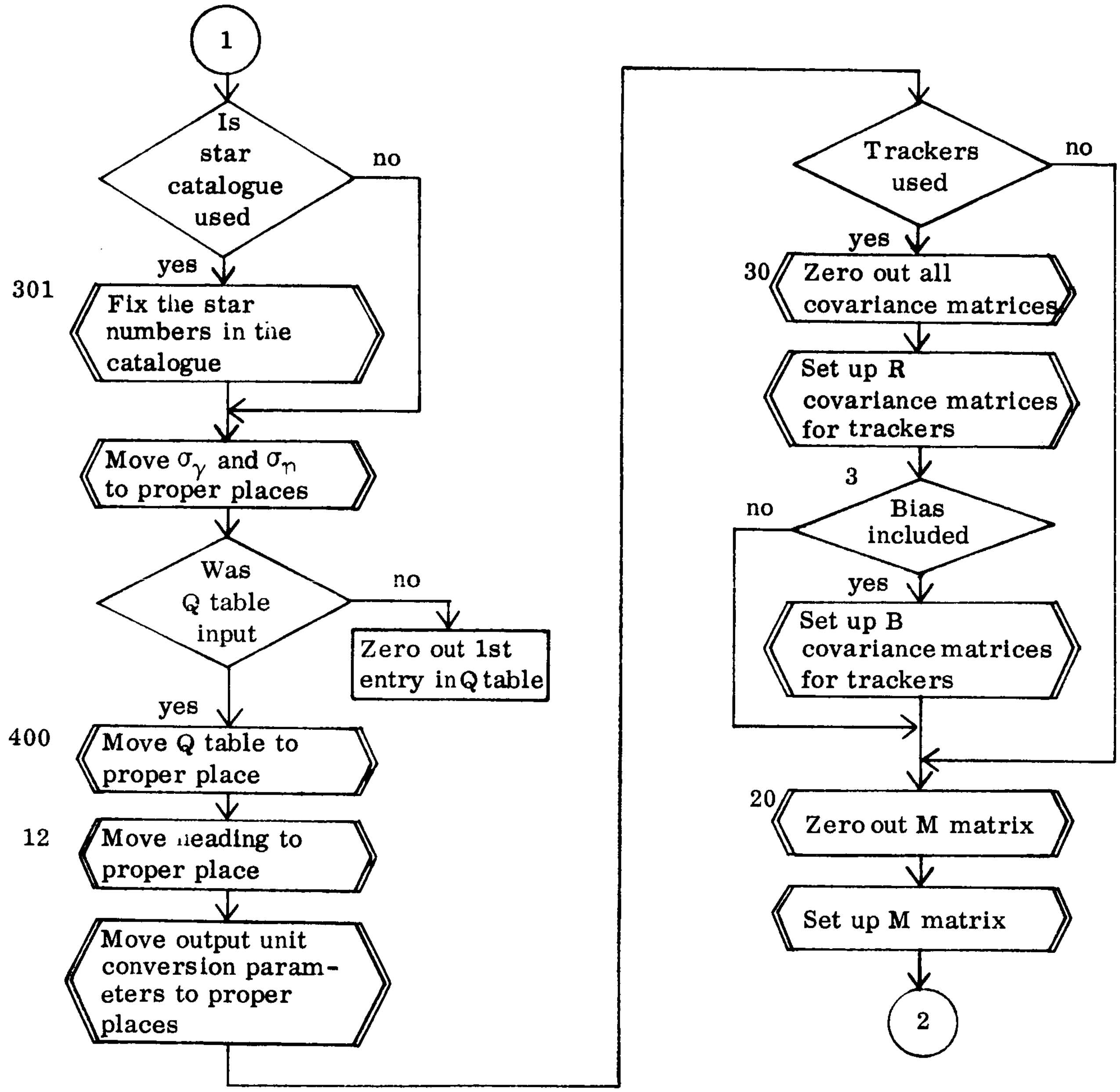
The following table indicates the storage allocation for the temporary input array T:

T (1) ↓ 20	Heading				
21	OUFG	40	$\sigma_{\dot{\rho}\eta 1}$	59	M_{66}
22	OUL	41	$\sigma_{\psi\eta 1}$	60	M_{12}
23	OUT	42	$\sigma_{\dot{\rho}\psi 1}$	61	M_{13}
24	$\sigma_{\psi 1}^{2}$	43	$\sigma_{\dot{\rho}\eta 2}$	62	M_{14}
25	$\sigma_{\eta 1}^{2}$	44	$\sigma_{\psi\eta 2}$	63	M_{15}
26	$\sigma_{\psi 2}^{2}$	45	$\sigma_{\dot{\rho}\psi 3}$	64	M_{16}
27	$\sigma_{\eta 2}^{2}$	46	$\sigma_{\dot{\rho}\eta 3}$	65	M_{23}
28	$\sigma_{\psi 3}^{2}$	47	$\sigma_{\psi\eta 3}$	66	M_{24}
29	$\sigma_{\eta 3}^{2}$	48	B4 (1, 1)	67	M_{25}
30	$\sigma_{\rho\dot{\rho}1}$	49	(2, 2)	68	M_{26}
31	$\sigma_{\rho\psi 1}$	50	(3, 3)	69	M_{34}
32	$\sigma_{\rho\eta 1}$	51	(1, 2)	70	M_{35}
33	$\sigma_{\rho\dot{\rho}2}$	52	(1, 3)	71	M_{36}
34	$\sigma_{\rho\psi 2}$	53	(2, 3)	72	M_{45}
35	$\sigma_{\rho\eta 2}$	54	M_{11}	73	M_{46}
36	$\sigma_{\rho\dot{\rho}3}$	55	M_{22}	74	M_{47}
37	$\sigma_{\rho\psi 3}$	56	M_{33}	75	B_{1L} 11
38	$\sigma_{\rho\eta 3}$	57	M_{44}	76	22
39	$\sigma_{\dot{\rho}\psi 1}$	58	M_{55}	77	B_{1L} 33

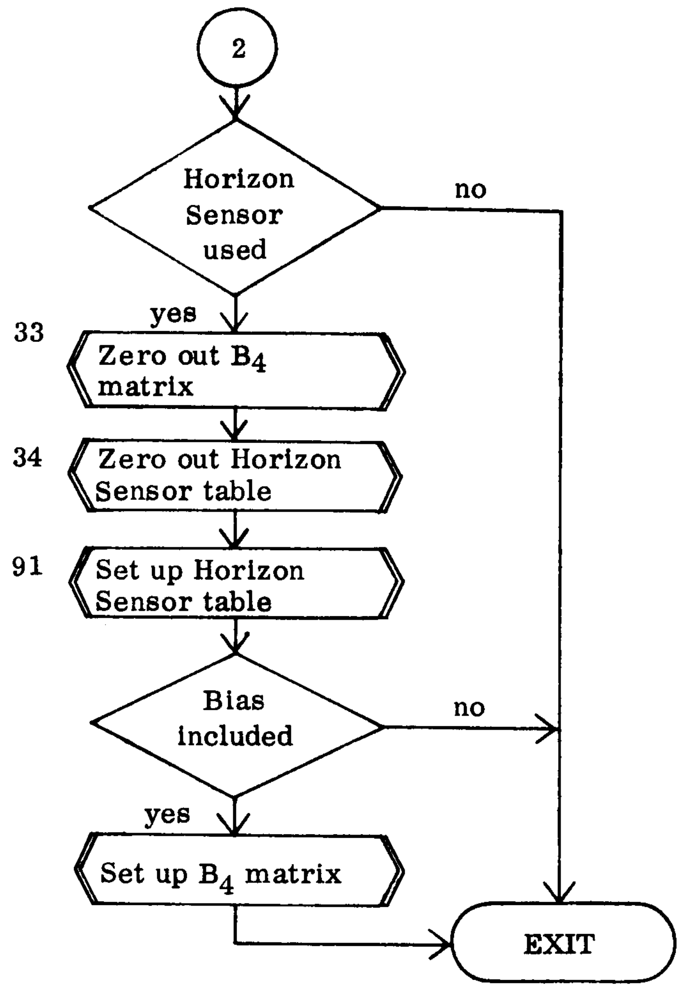
78	B_{2L}	11	101	B_{3I}	11	123	t_1
79		22	102		22	124	R^4_{12}
80		33	103		33	125	R^4_{13}
81	B_{3L}	11	104		44		t_2
82		22	105	B_{1I}	12		$\downarrow$
83		33	106		13	322	$\downarrow$
84	B_{1L}	12	107		14	323	t_1
85		13	108	B_{2I}	12		R^4_{11}
86		23	109		13		R^4_{22}
87	B_{2L}	12	110		14		R^4_{33}
88		13	111	B_{3I}	12		t_2
89		23	112		13		$\downarrow$
90	B_{3L}	12	113		14	522	$\downarrow$
91		13	114	B_{1I}	23	523	σ^{γ}
92		23	115		24	524	σ^{η}
93	B_{1I}	11	116		34	525	t_1
94		22	117	B_{2I}	23	526	q_{11}
95		33	118		24	527	q_{22}
96		44	119		34	528	q_{33}
97	B_{2I}	11	120	B_{3I}	23	529	q_{44}
98		22	121		24	530	q_{55}
99		33	122		34	531	q_{66}
100		44				532	t_2
						594	$\downarrow$



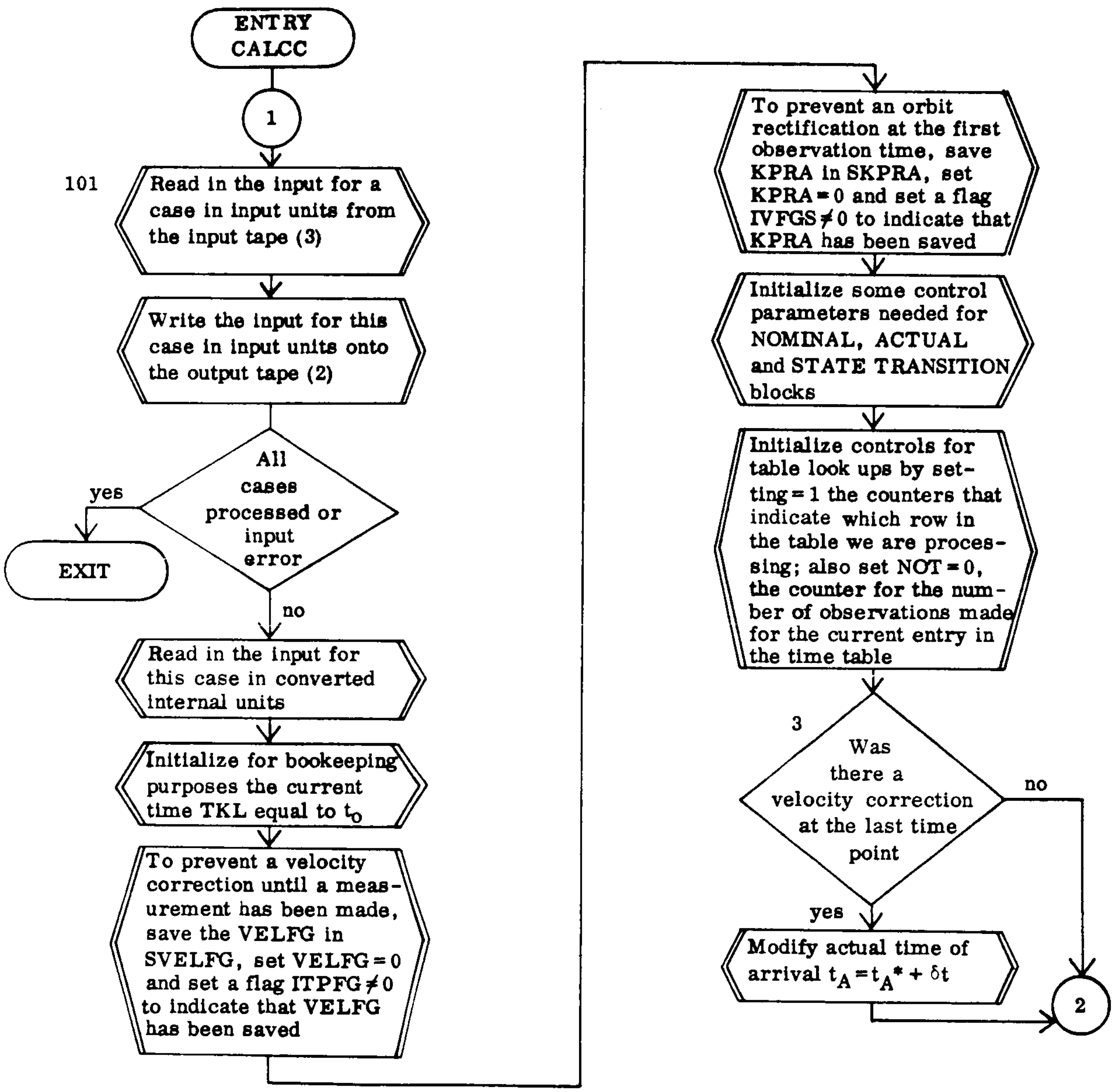
IMOVE - Input Data Preparation Subroutine



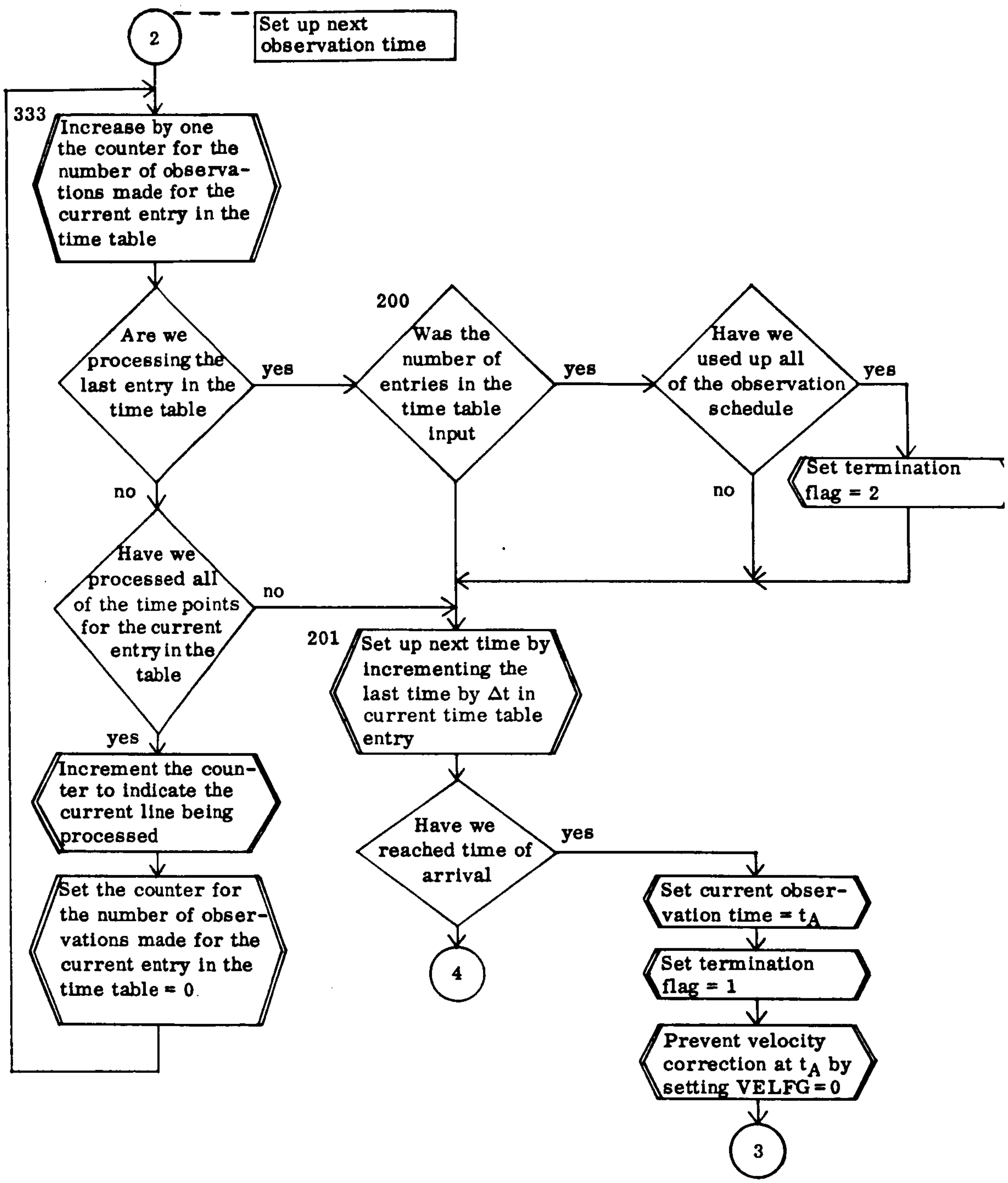
IMOVE - Input Data Preparation Subroutine (contd)



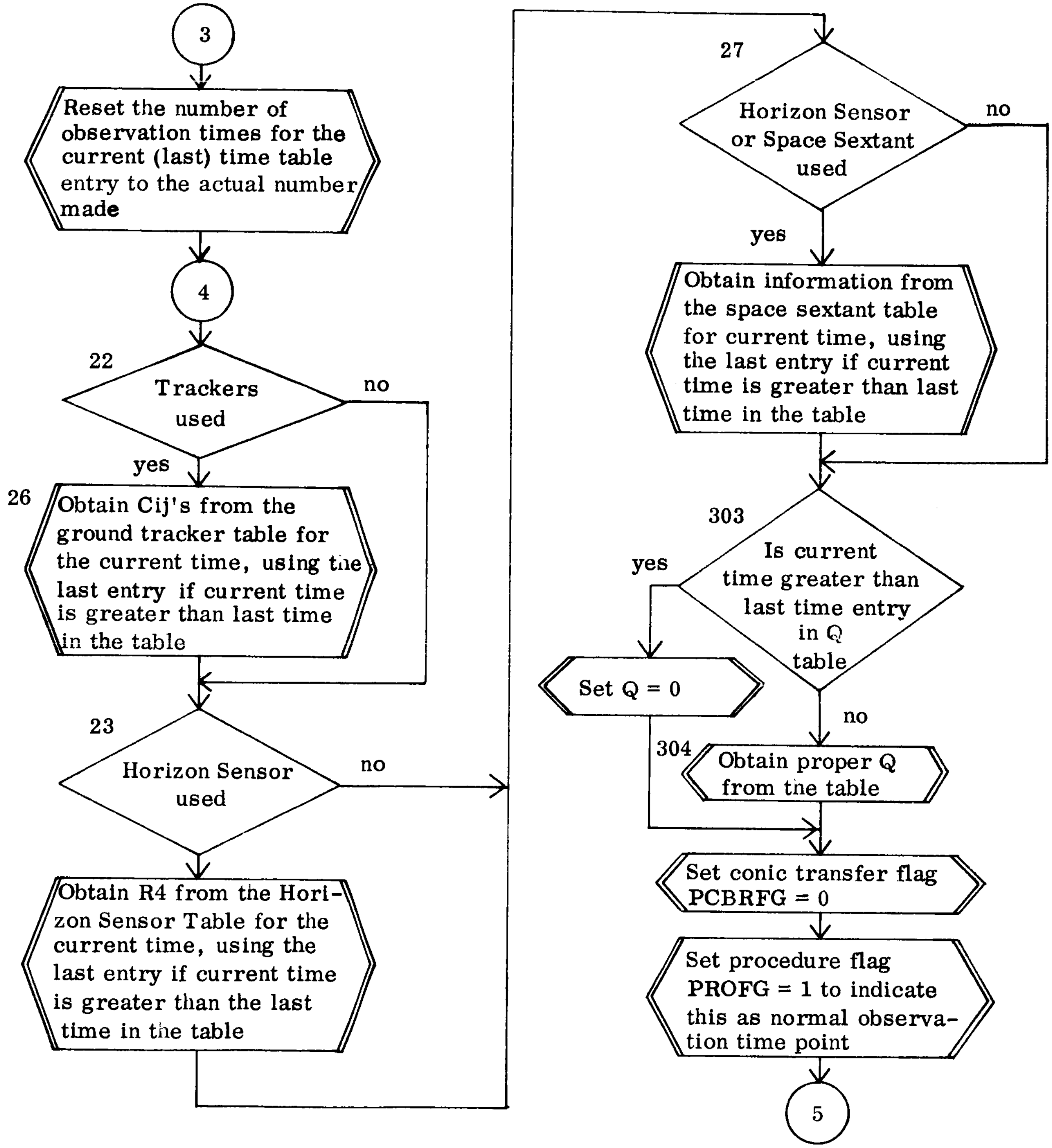
IMOVE - Input Data Preparation Subroutine (contd)



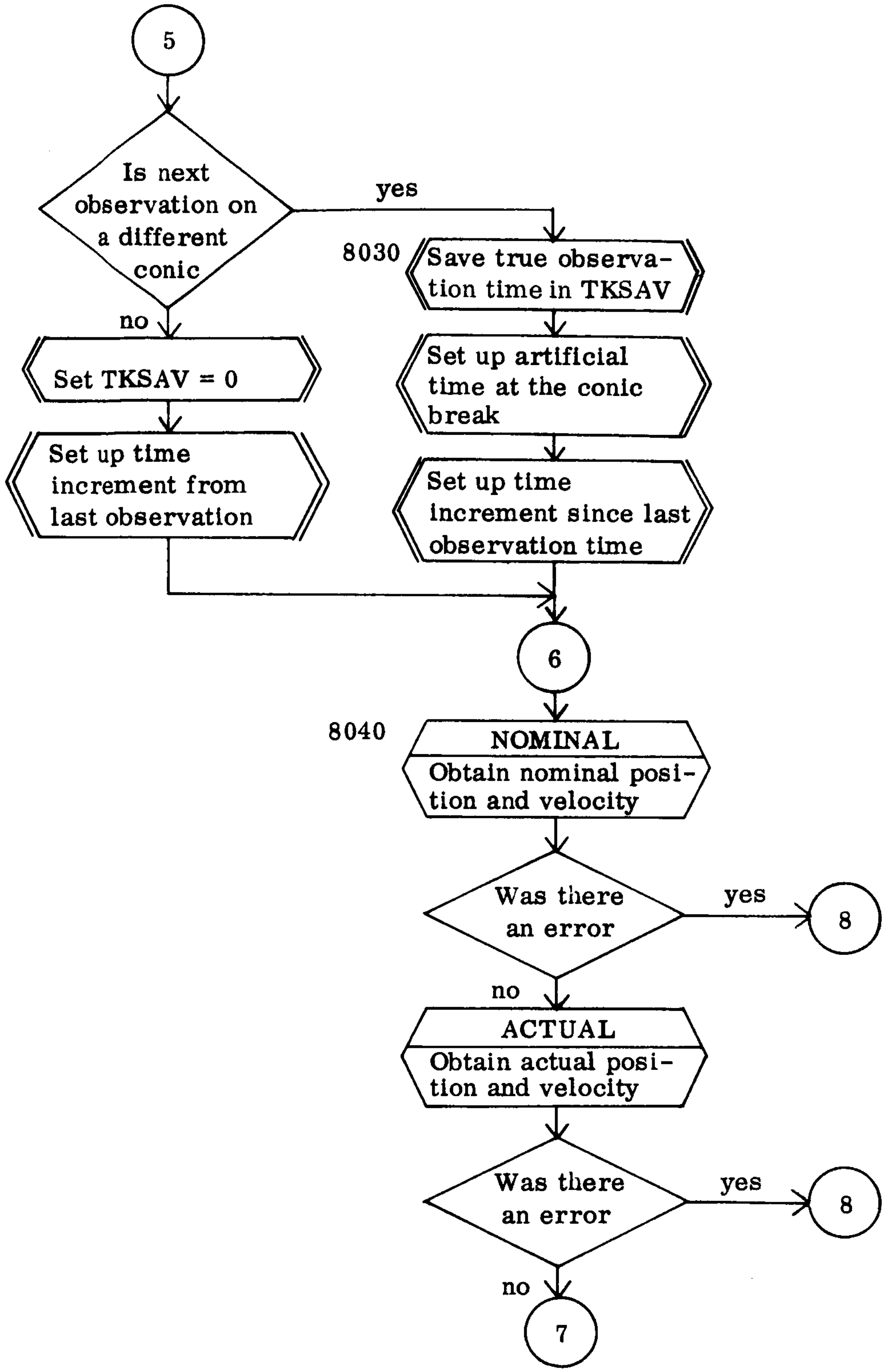
6.4.3 CALCC - Calculate Section Control Program



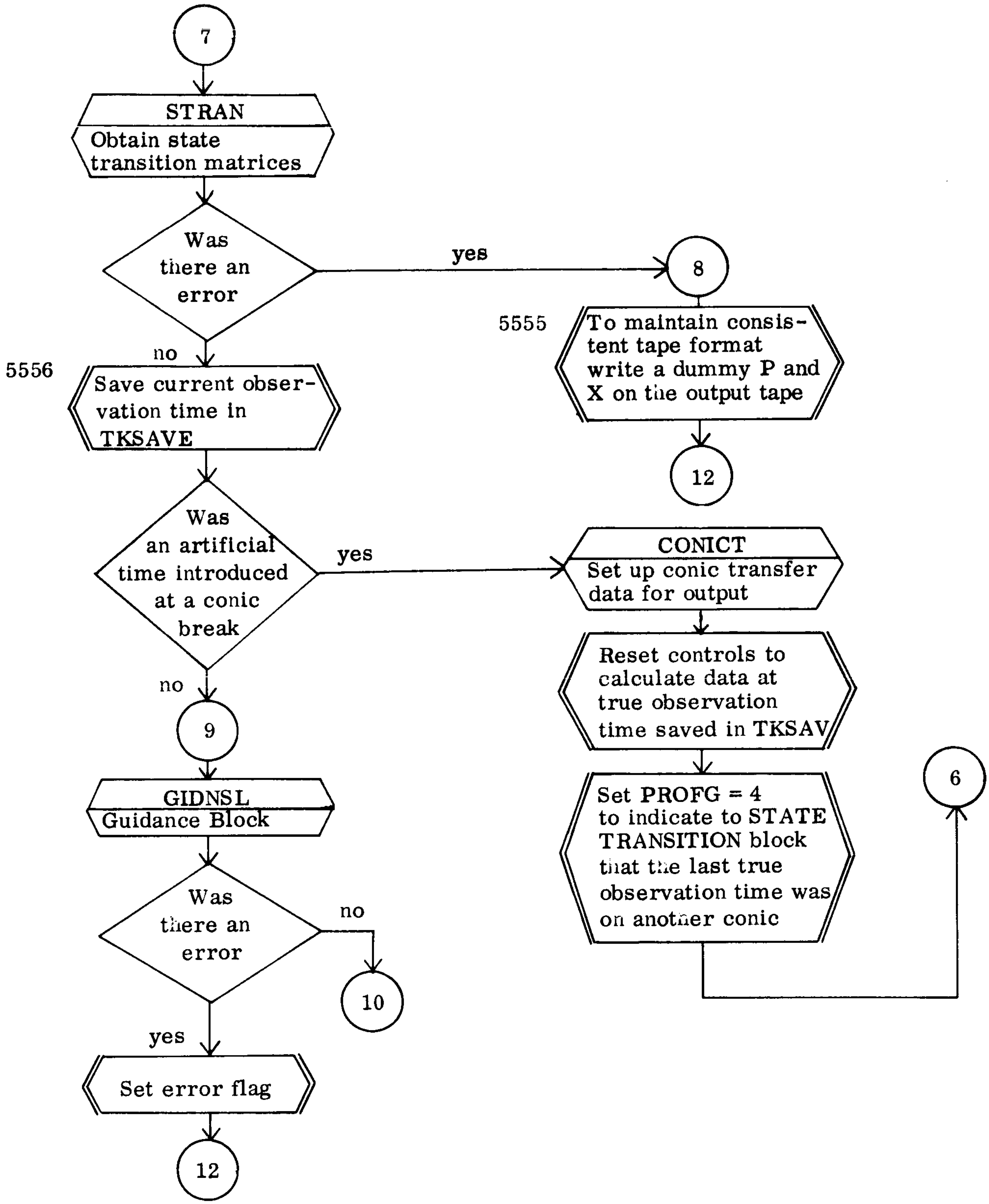
CALCC - Calculate Section Control Program (contd)



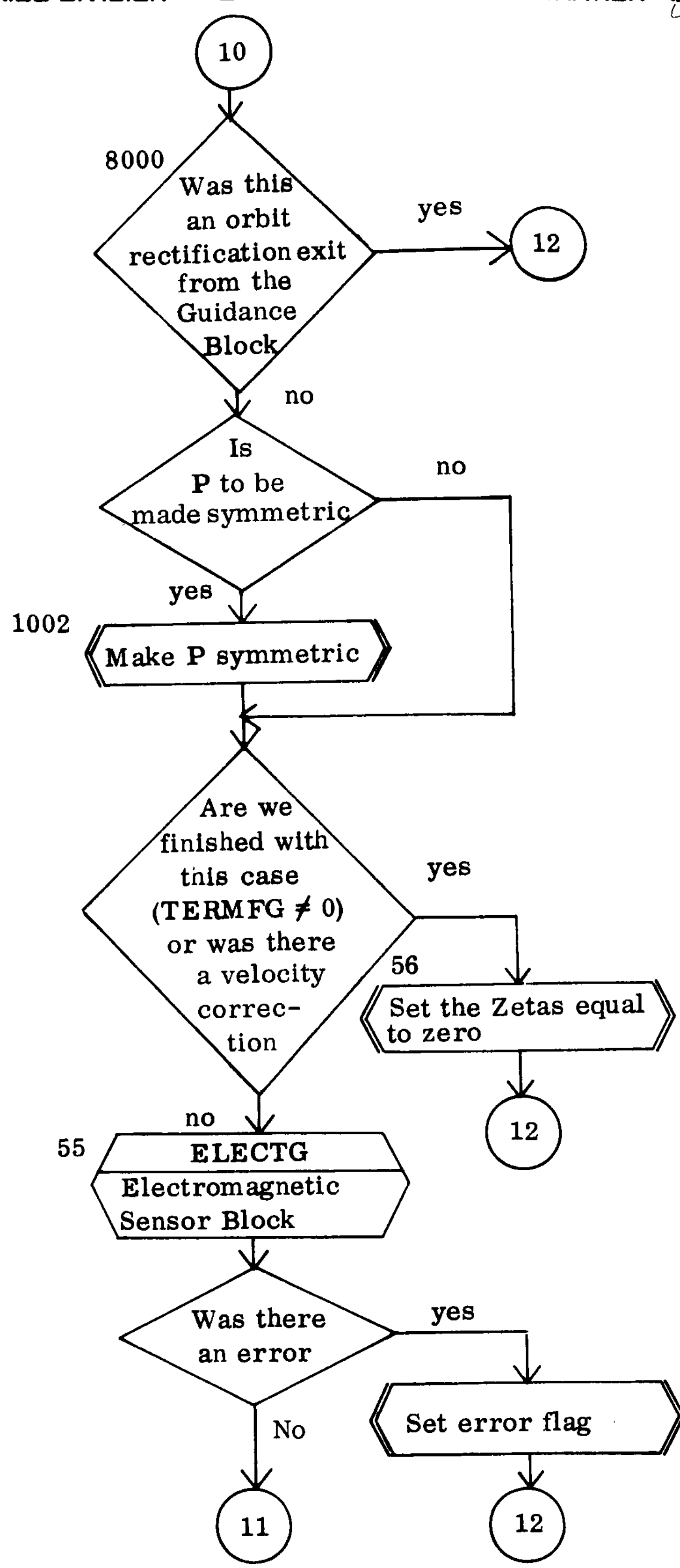
CALCC - Calculate Section Control Program (contd)



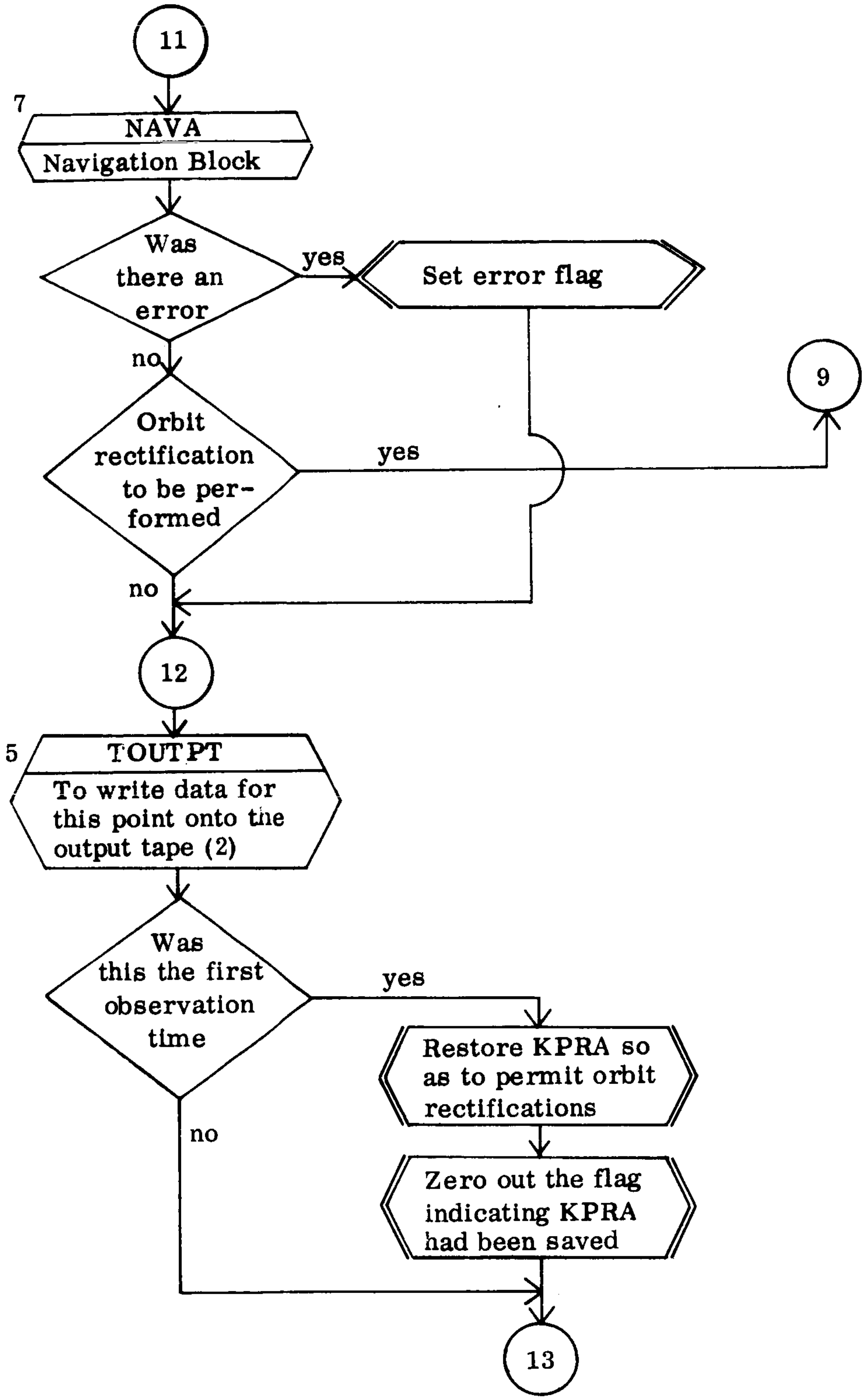
CALCC - Calculate Section Control Program (contd)



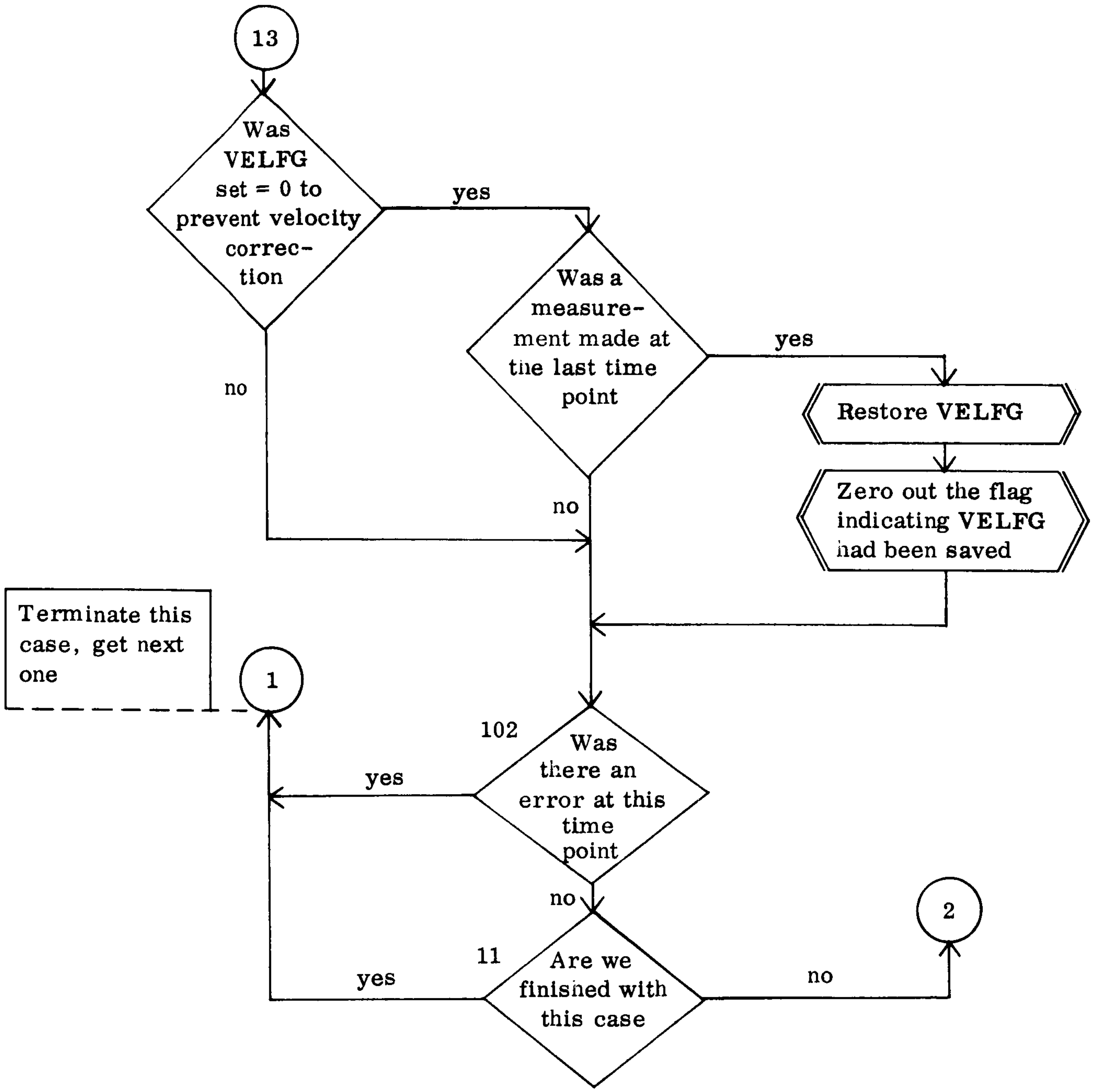
CALCC - Calculate Section Control Program (contd)



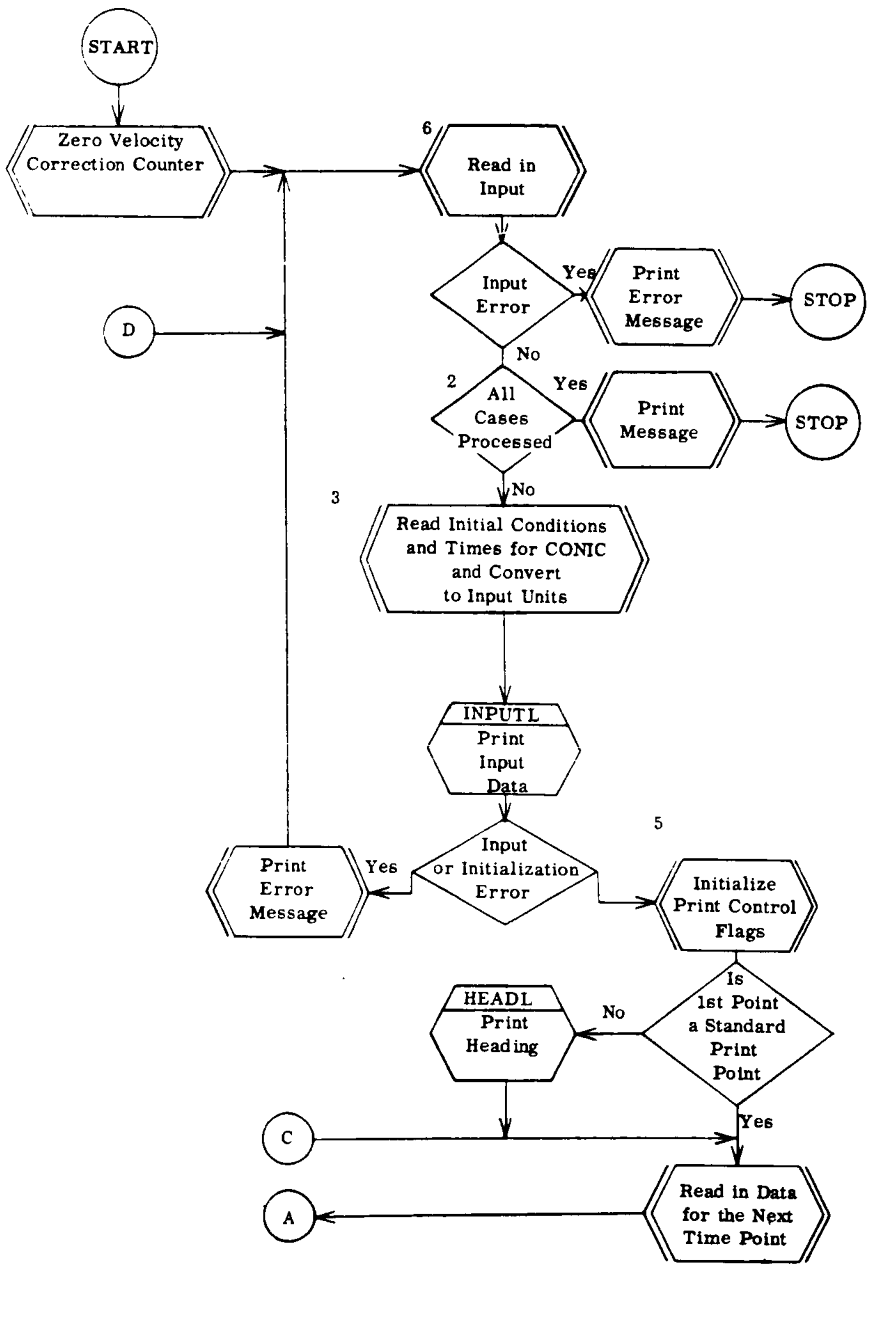
CALCC - Calculate Section Control Program (contd)



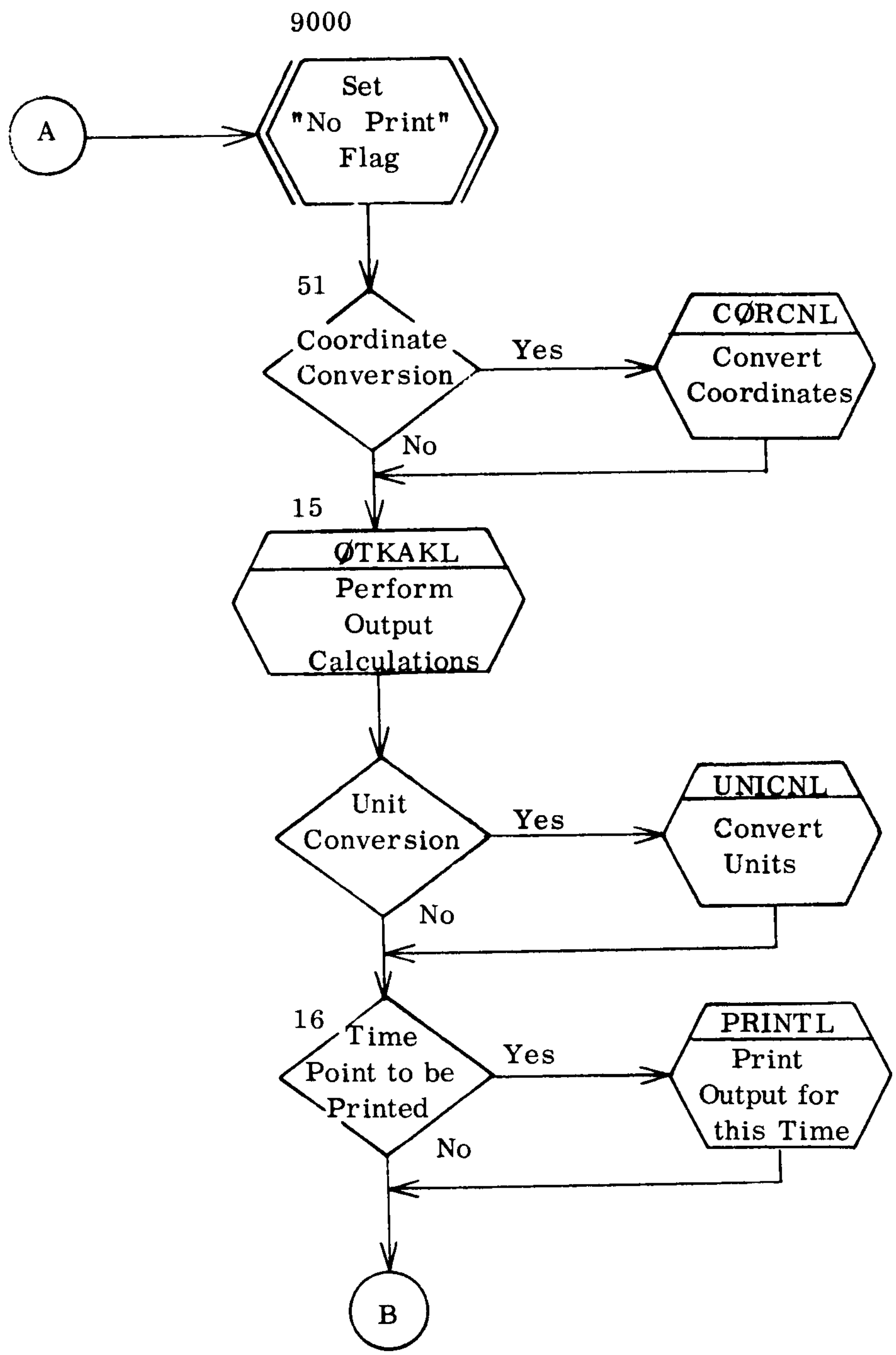
CALCC - Calculate Section Control Program (contd)



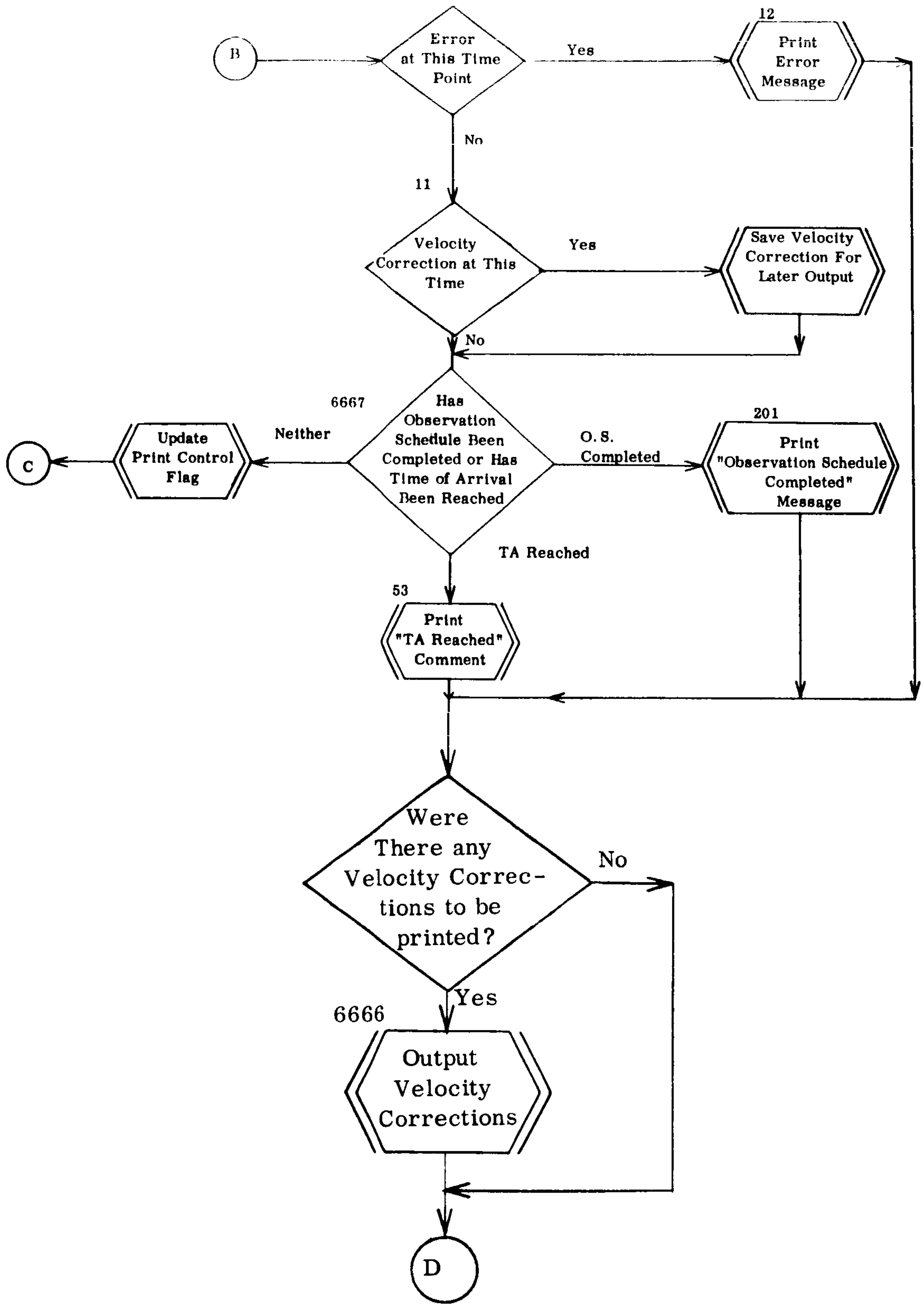
CALCC - Calculate Section Control Program (contd)



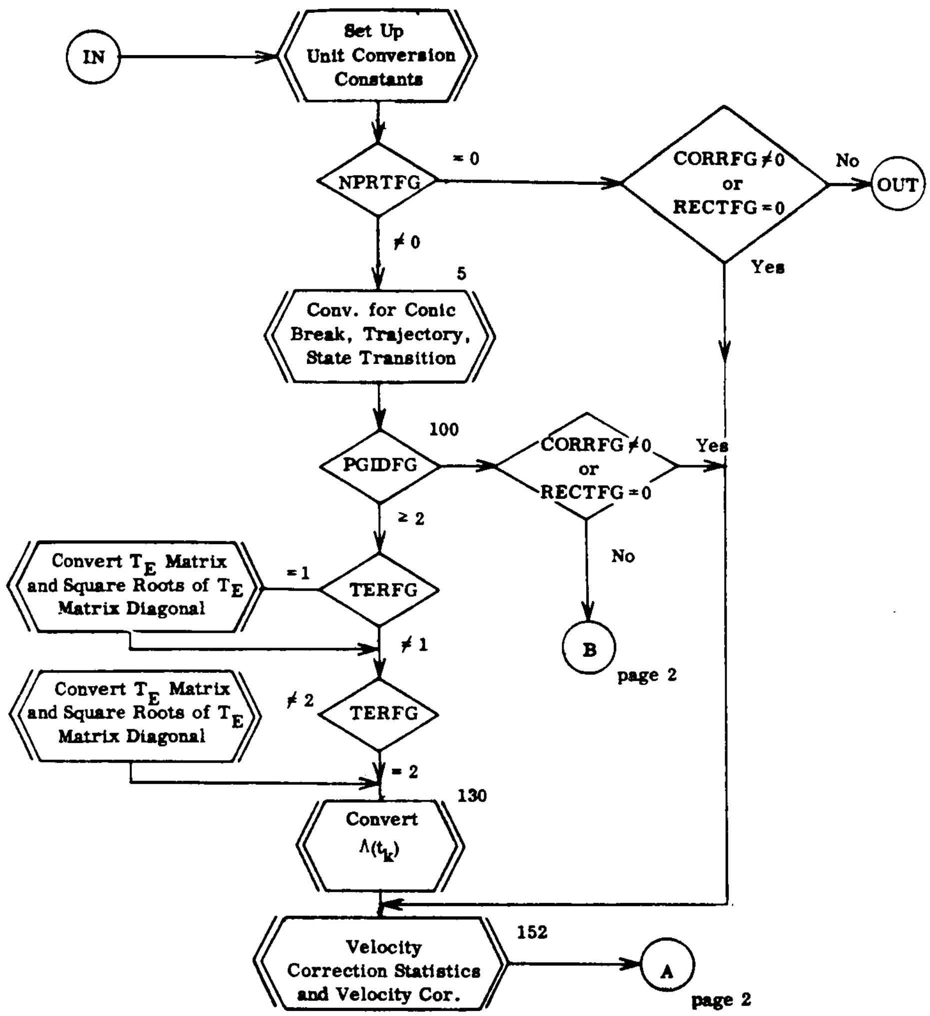
6.4.4 OUTPUT - Output Section Control Program



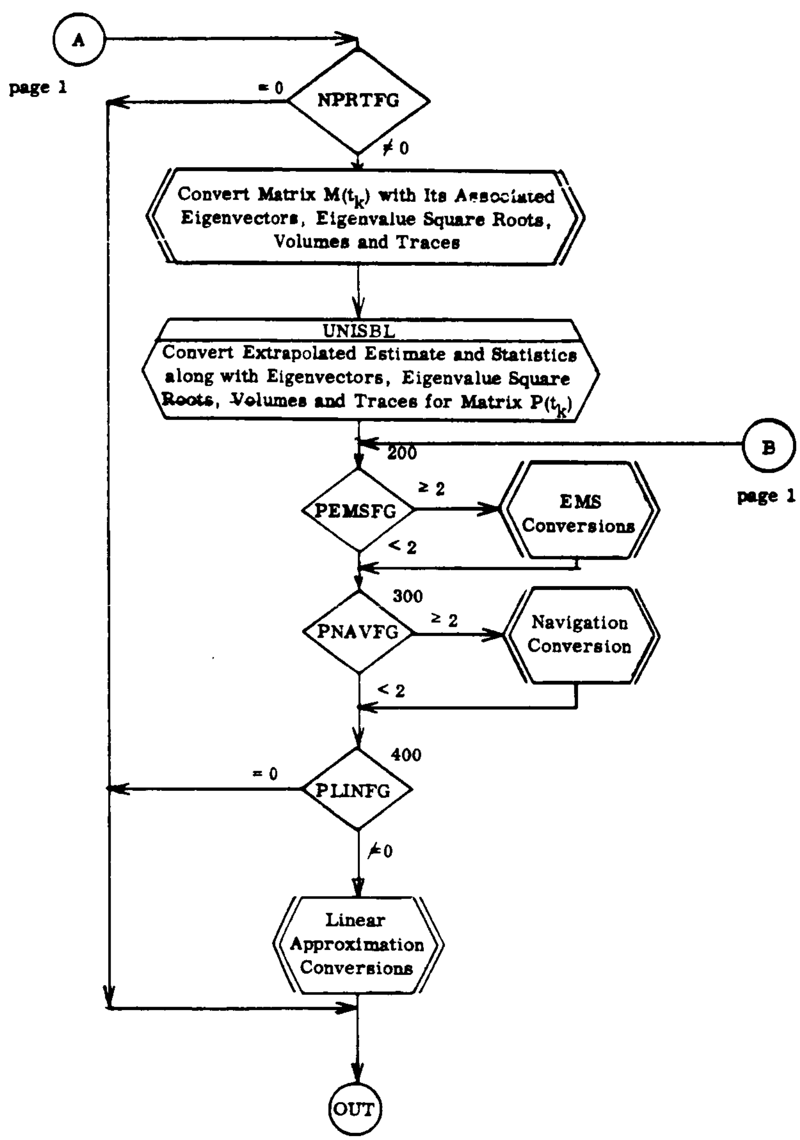
OUTPUT - Output Section Control Program (contd)



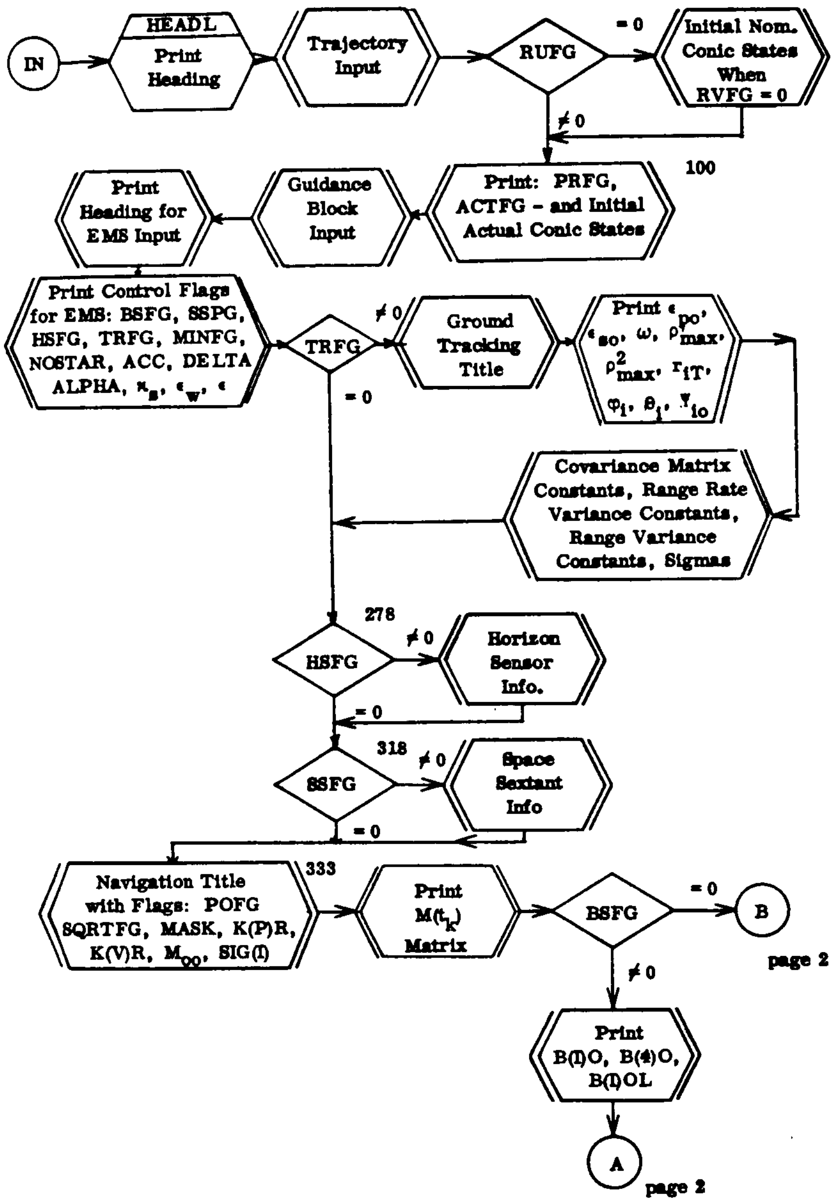
OUTPUT - Output Section Control Program (contd)



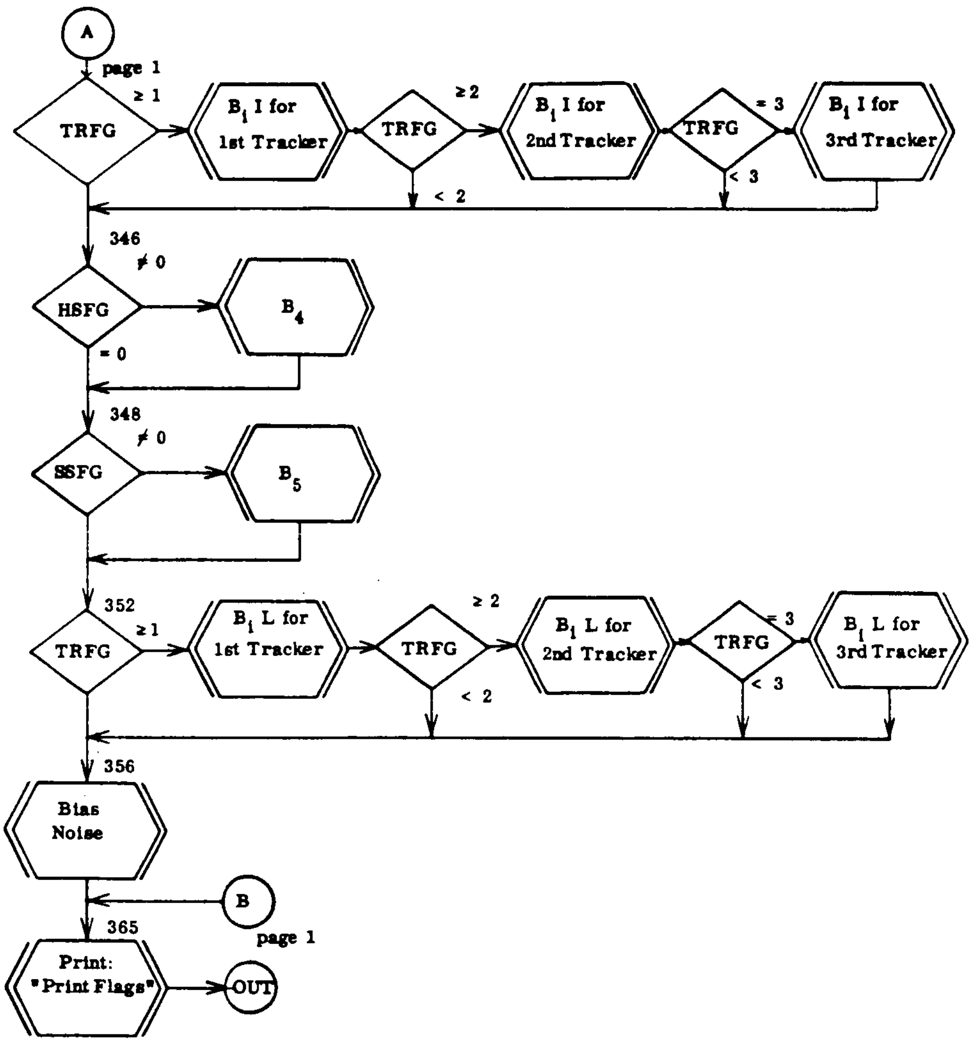
6.4.4.1 UNICNL - Subroutine



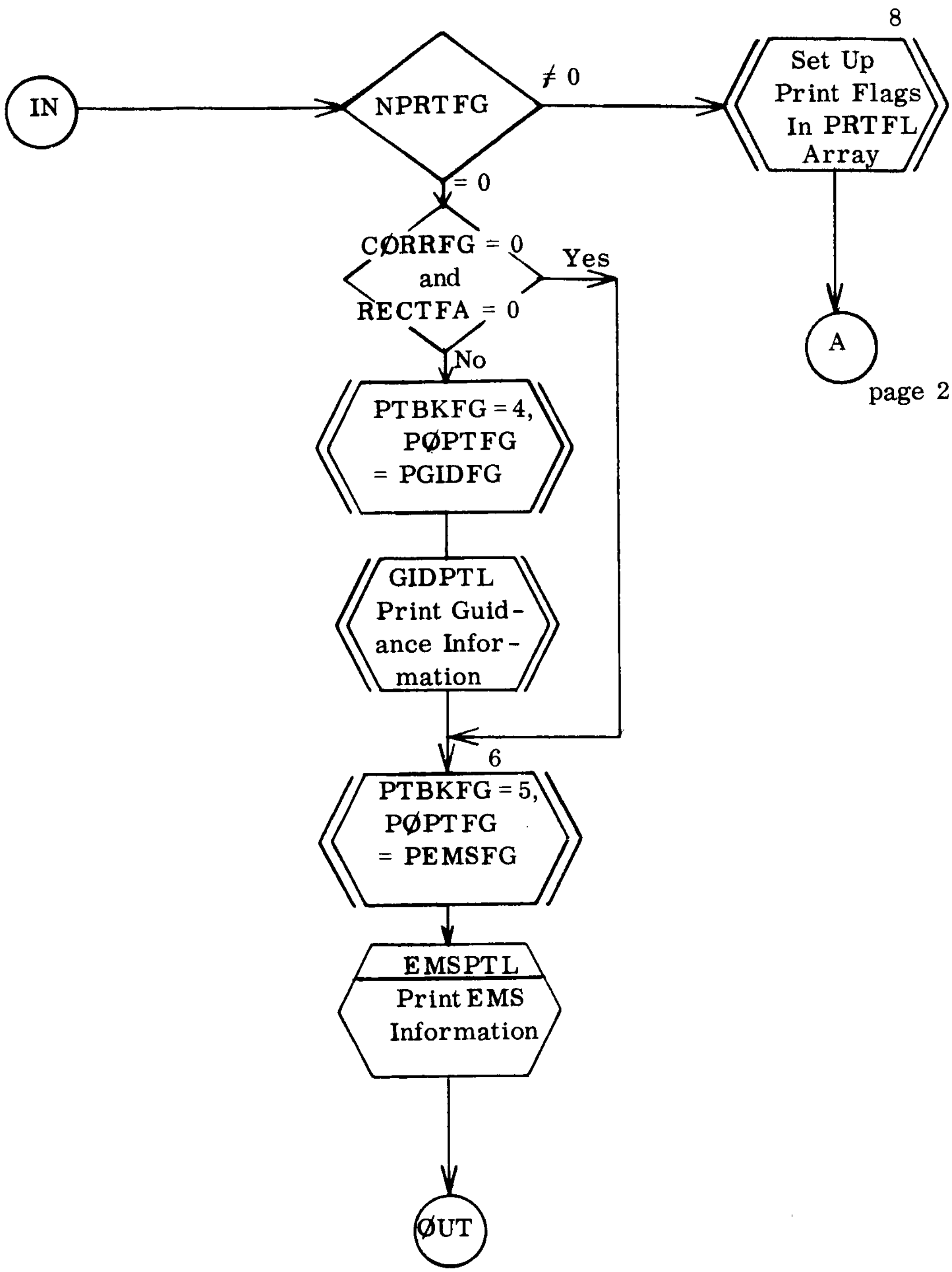
UNICNL - Subroutine (contd)



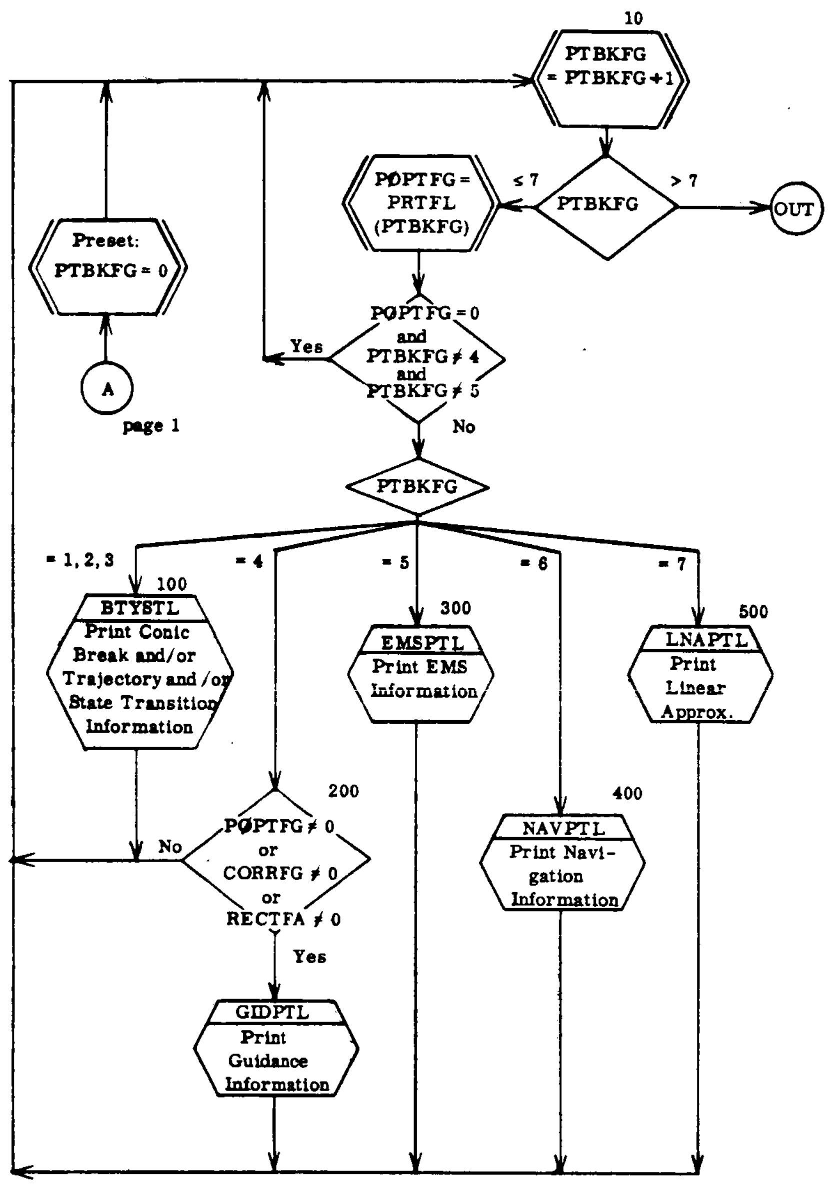
6.4.4.2 INPUTL - Subroutine



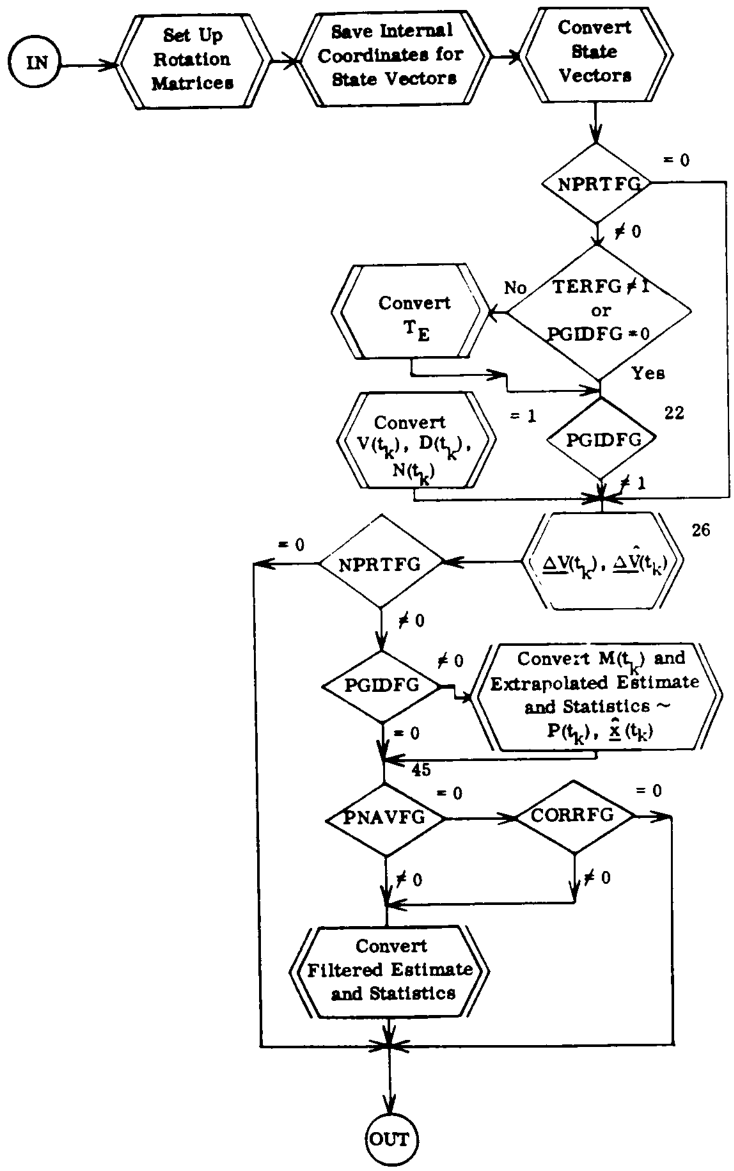
INPUTL - Subroutine (contd)



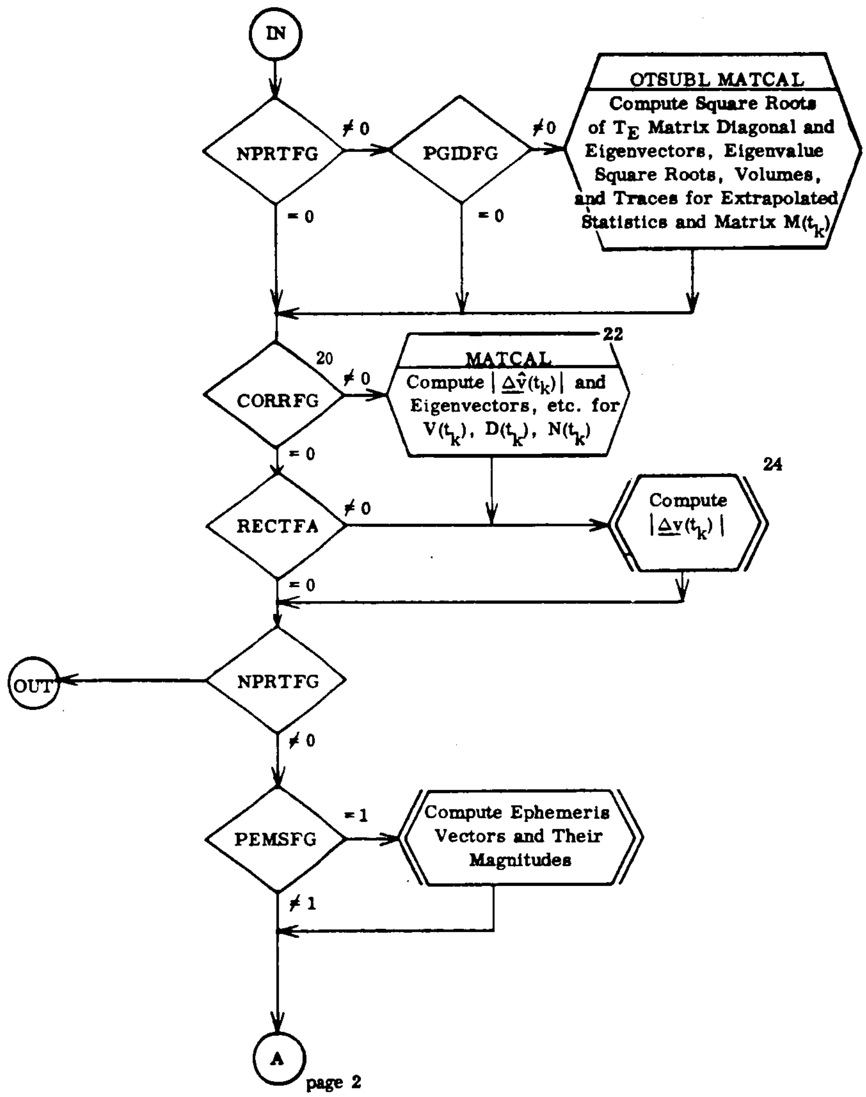
6.4.4.3 PRINTL - Subroutine



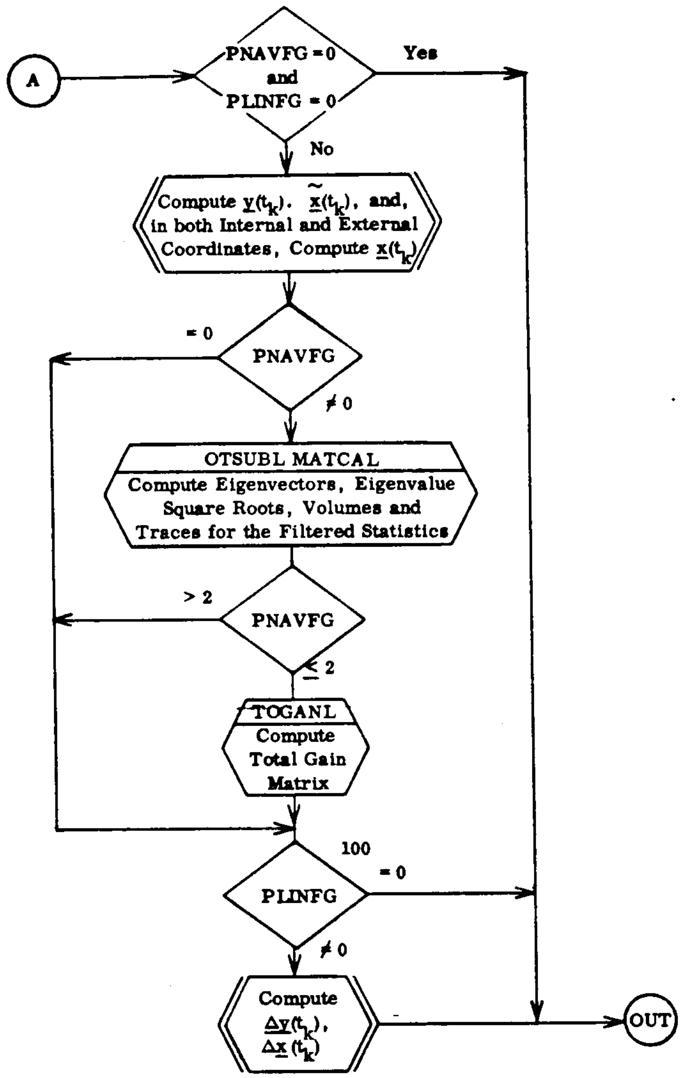
PRINTL - Subroutine (contd)



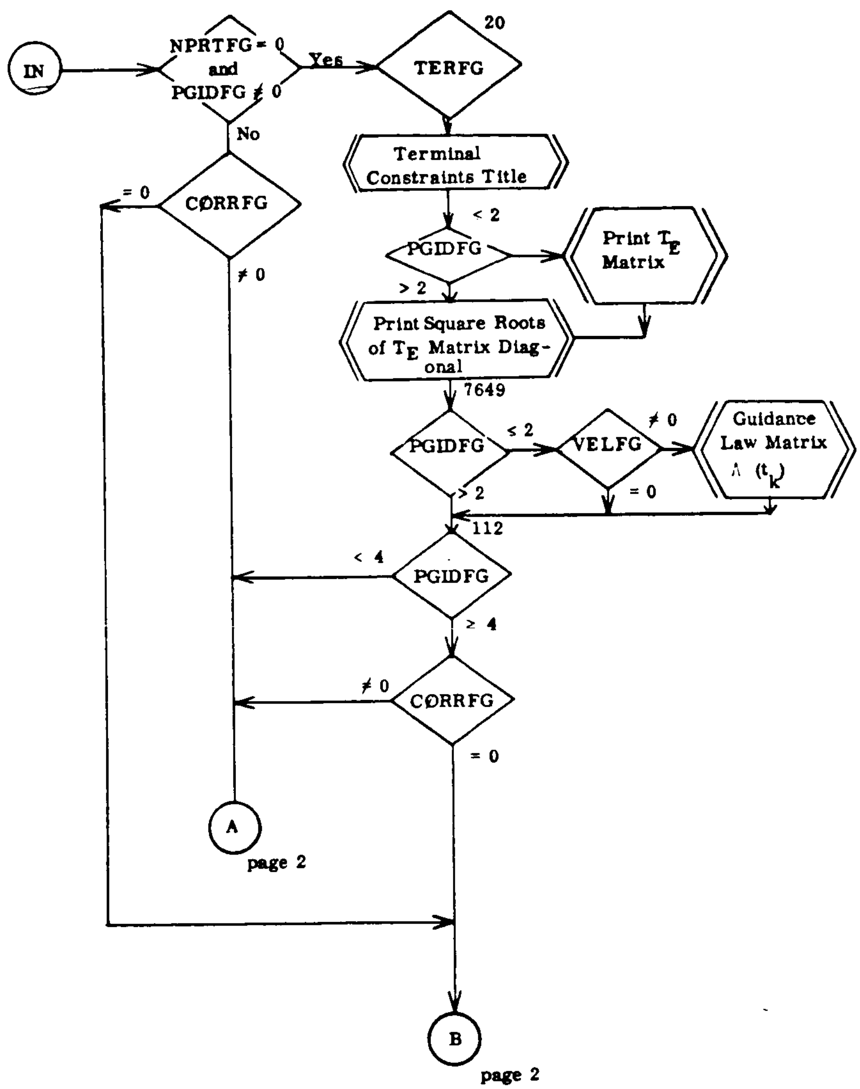
6.4.4.4 CORCNL - Subroutine



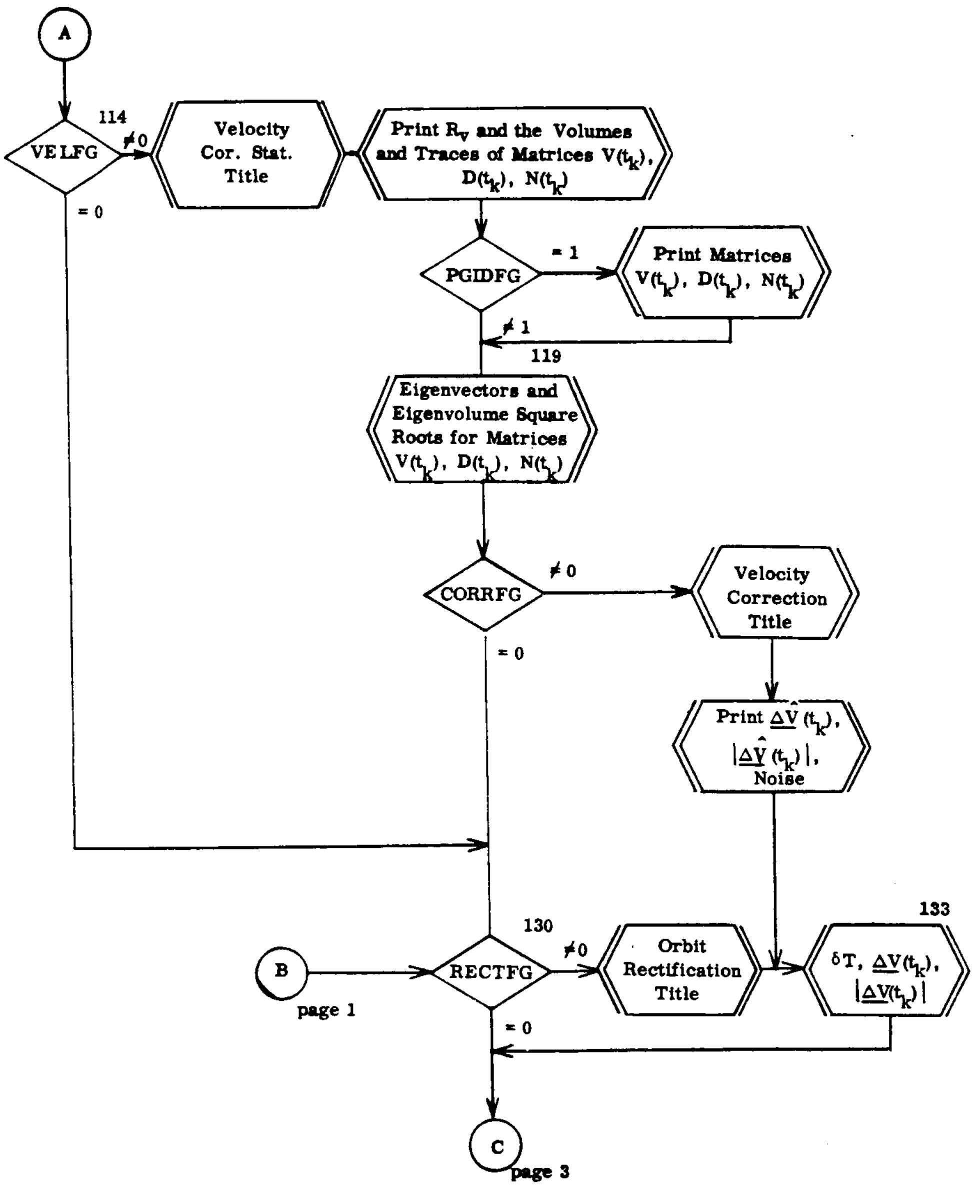
6.4.4.5 OTKAKL - Subroutine



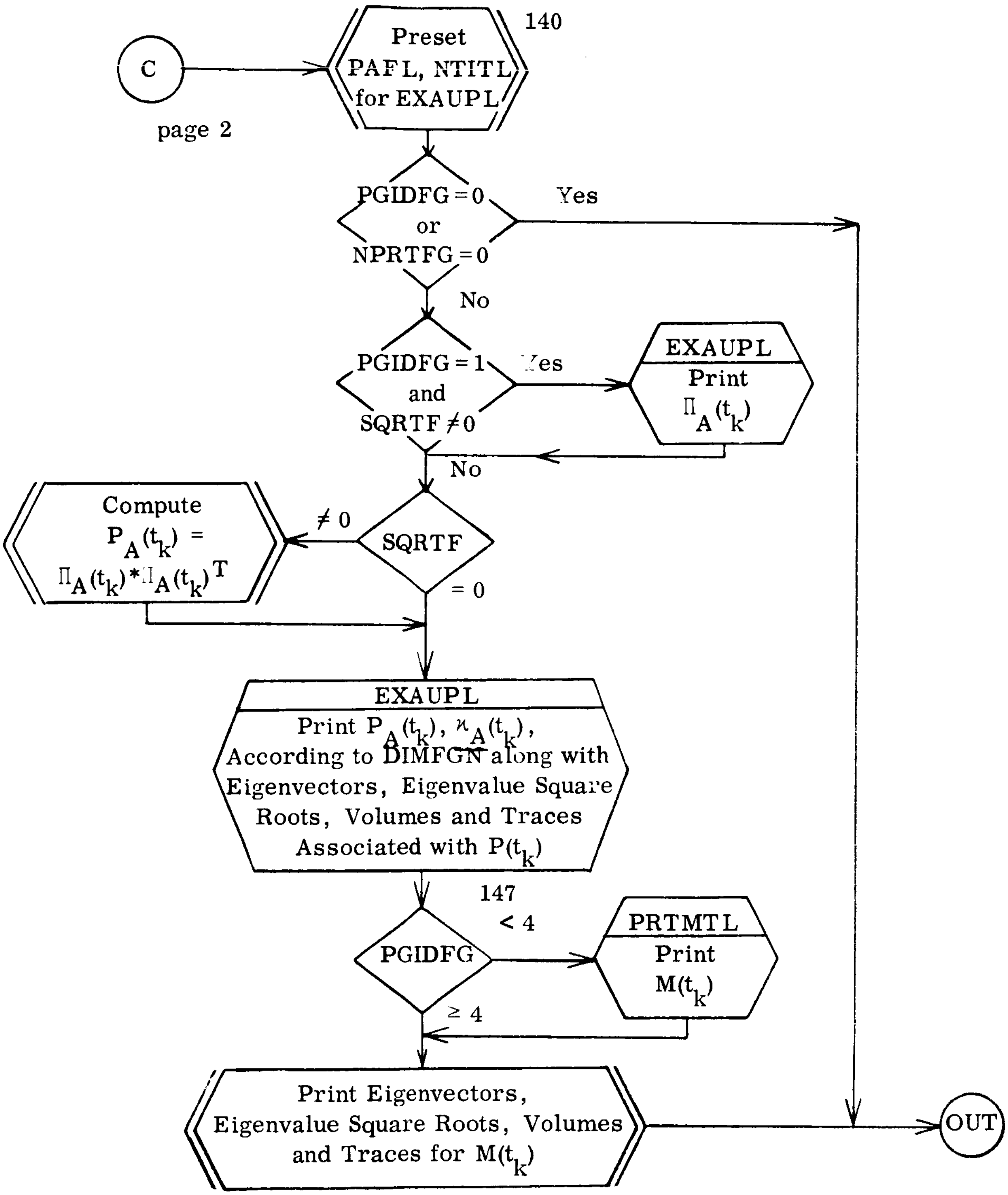
OTKAKL - Subroutine (contd)



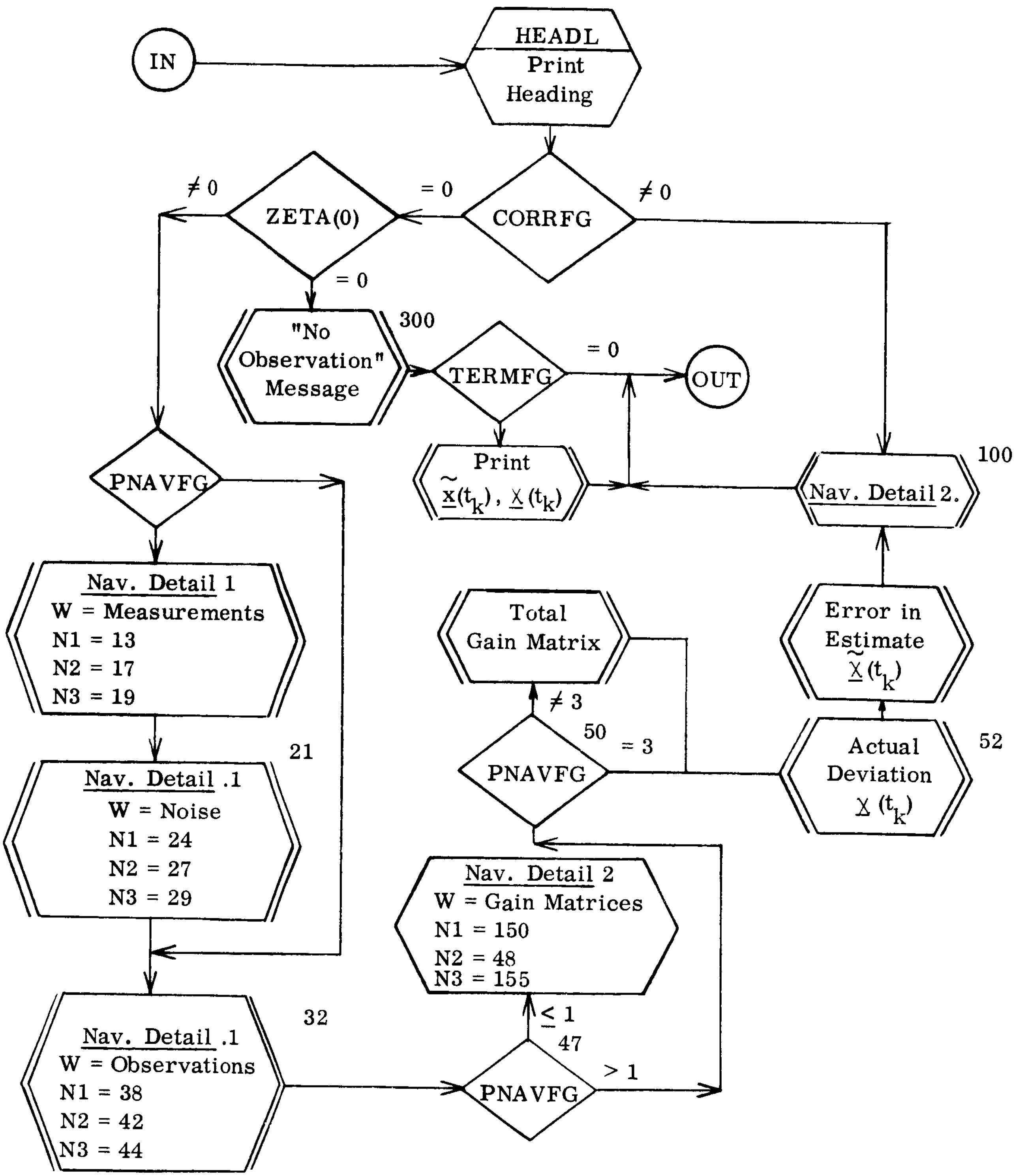
6.4.4.6 GIDPTL - Subroutine



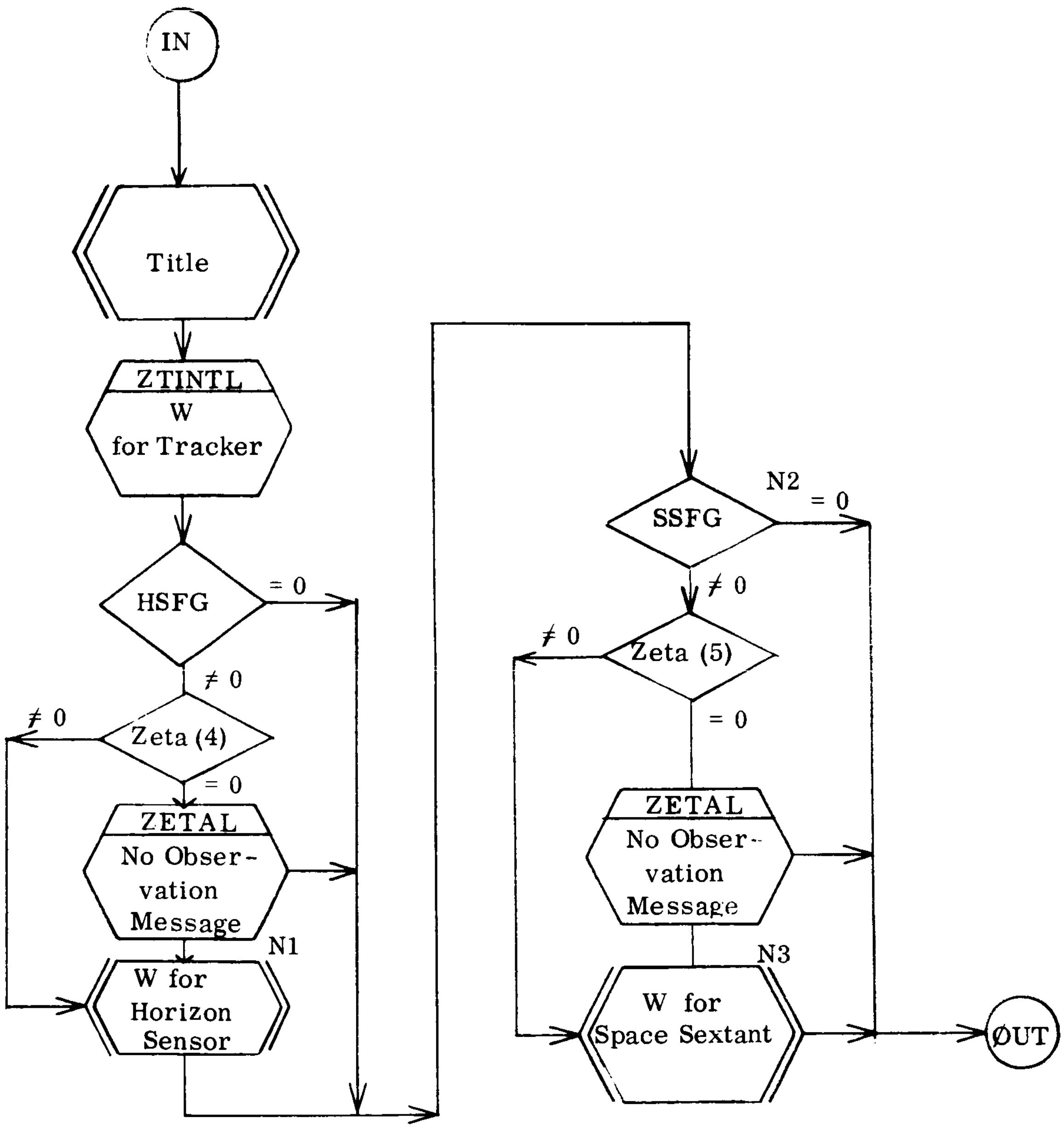
GIDPTL - Subroutine (contd)



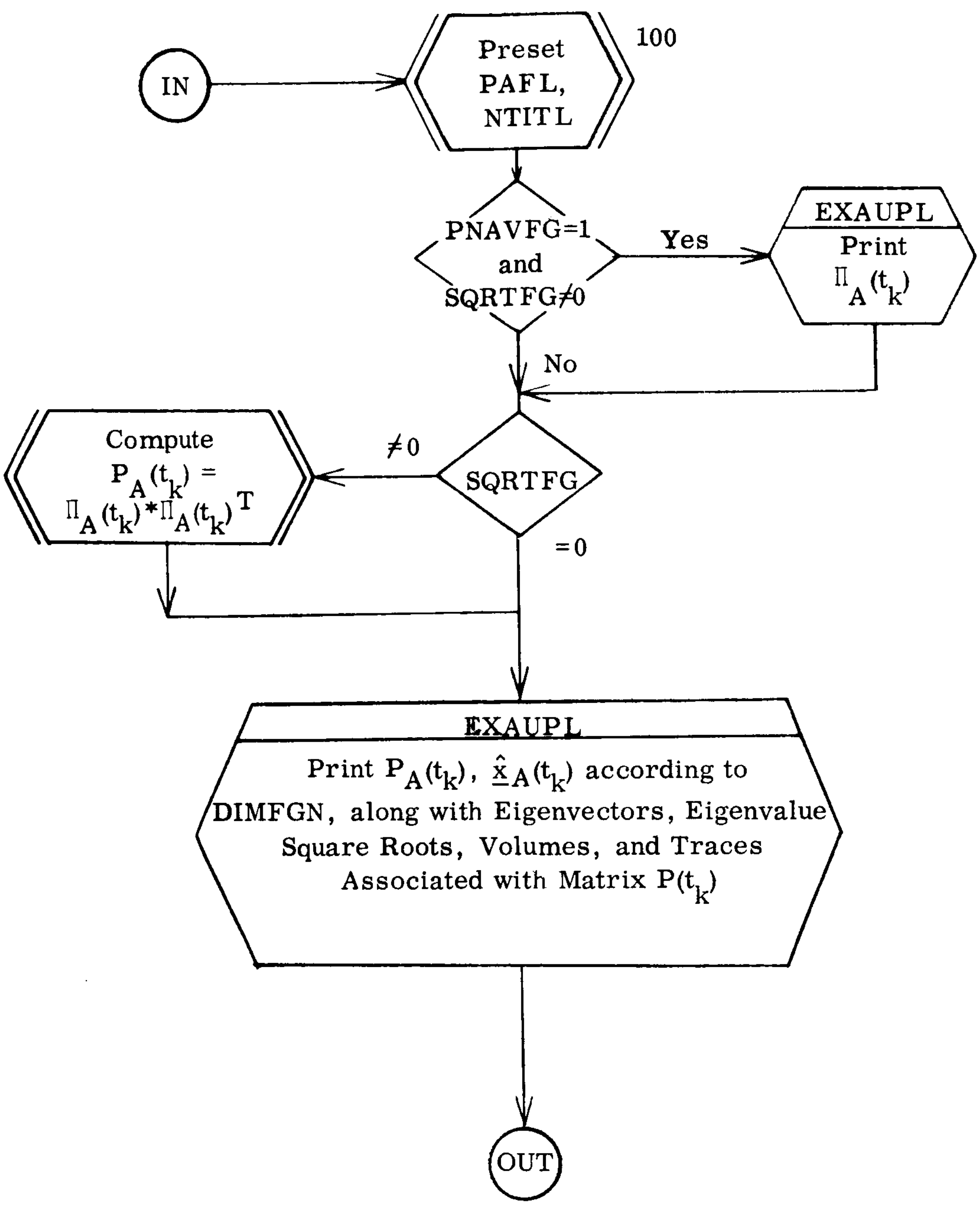
GIDPTL - Subroutine (contd)



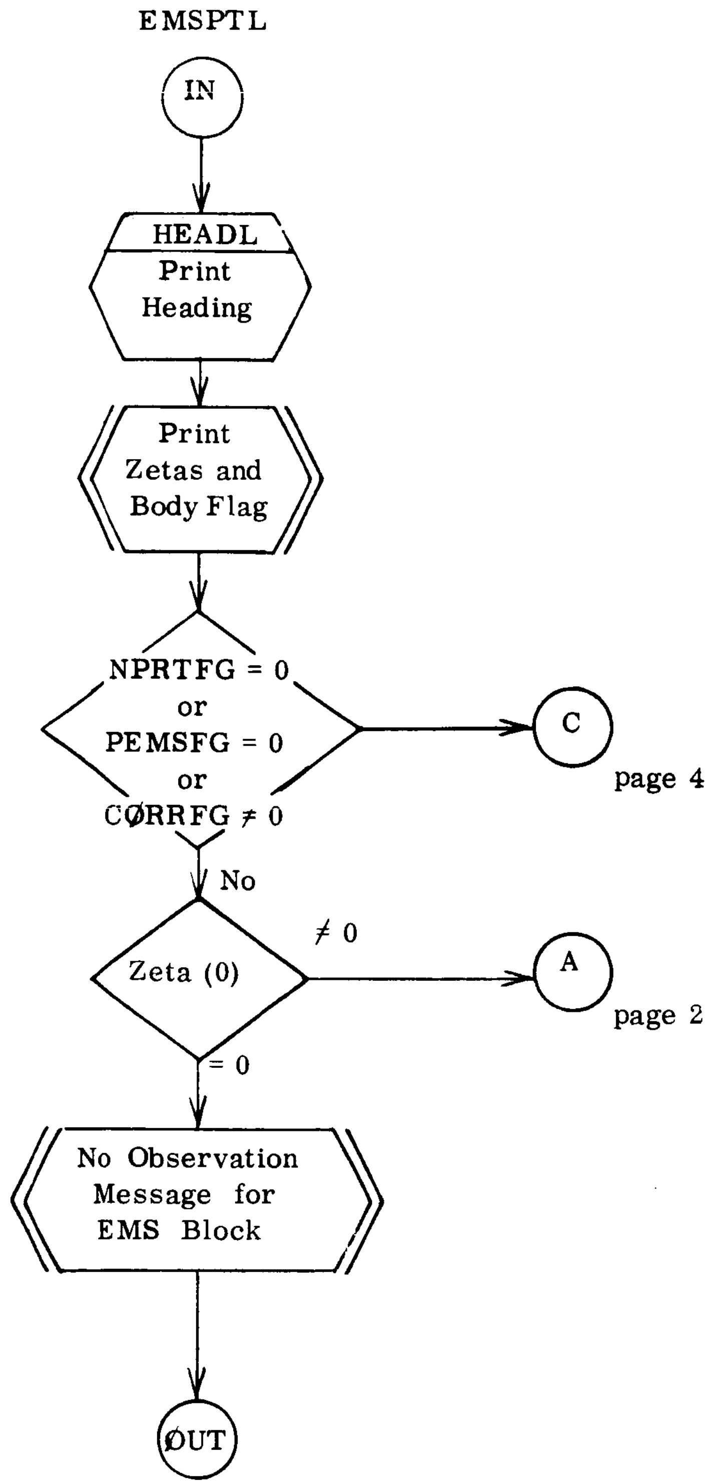
6.4.4.7 NAVPTL - Subroutine



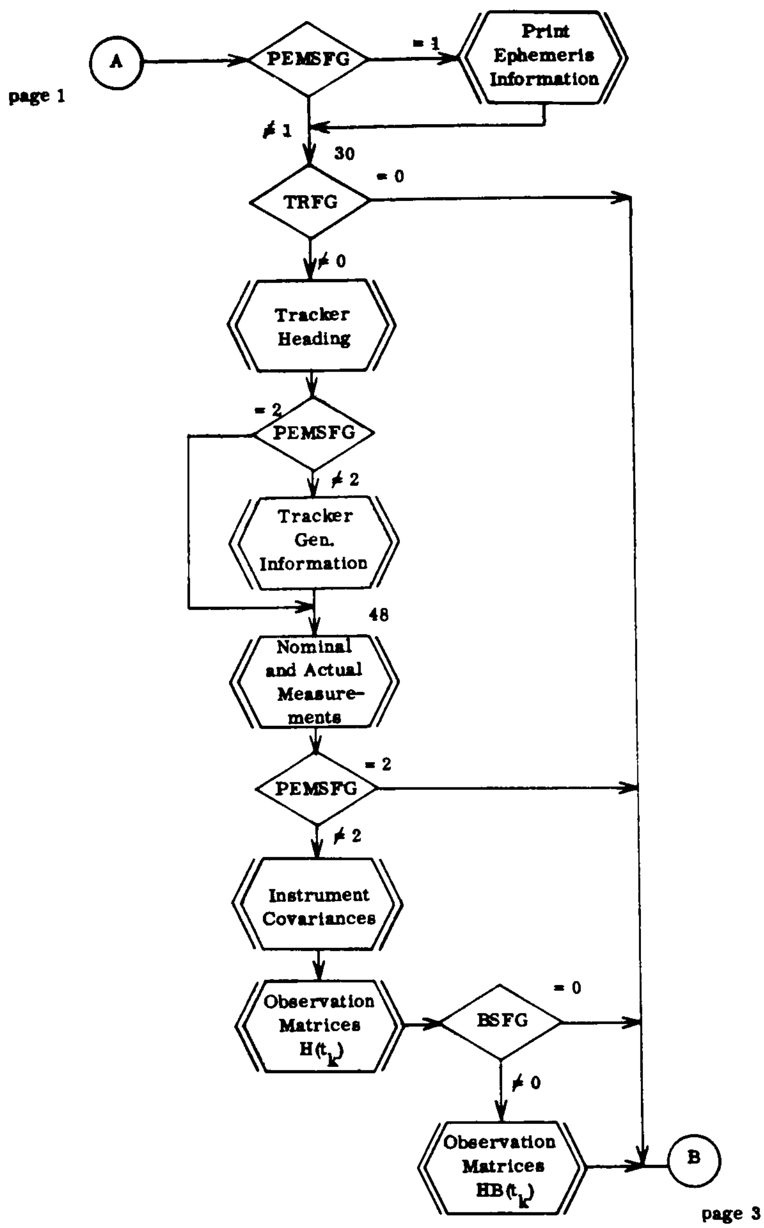
NAVPTL - Navigation Detail 1



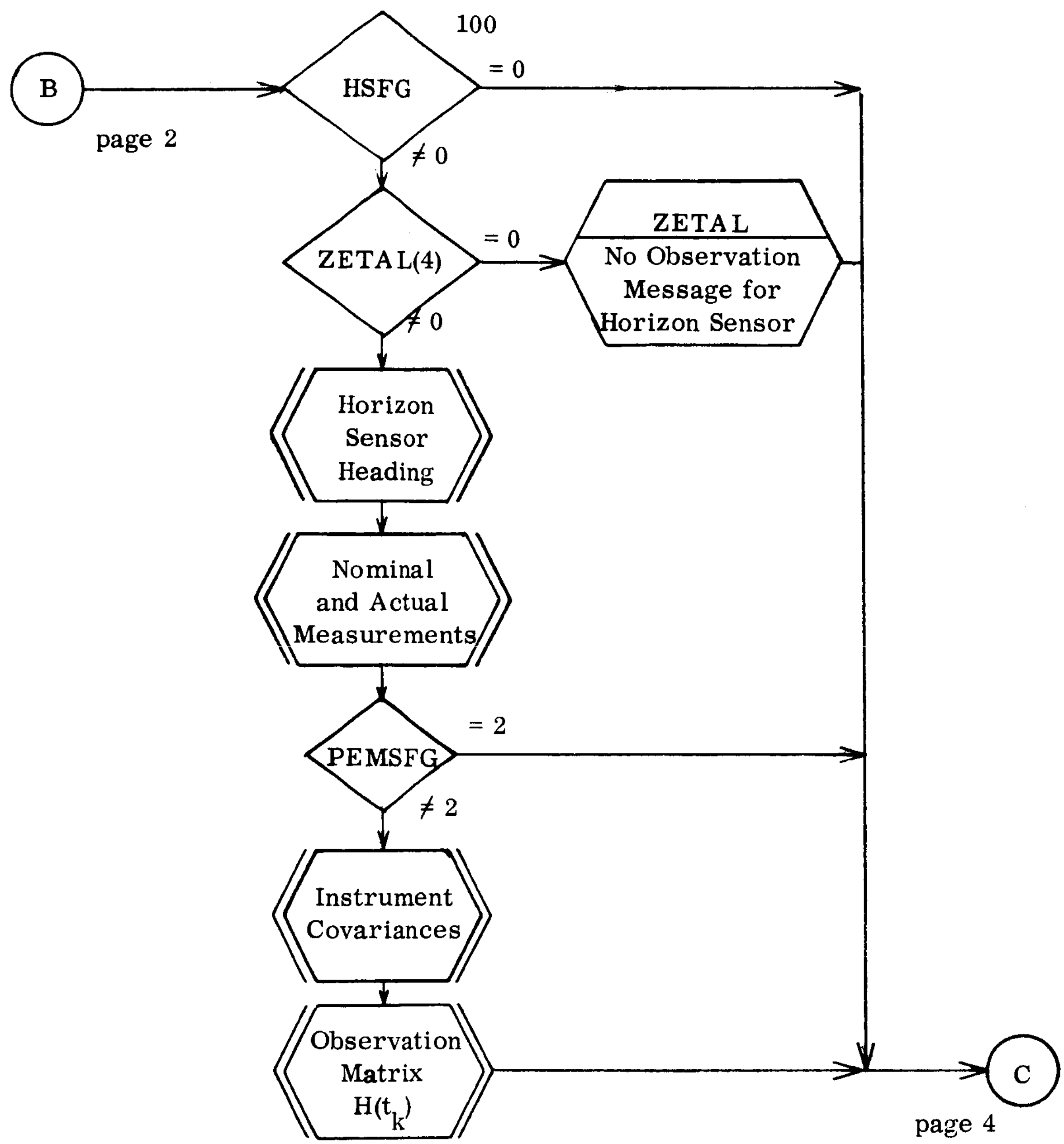
NAVPTL - Navigation Detail 2



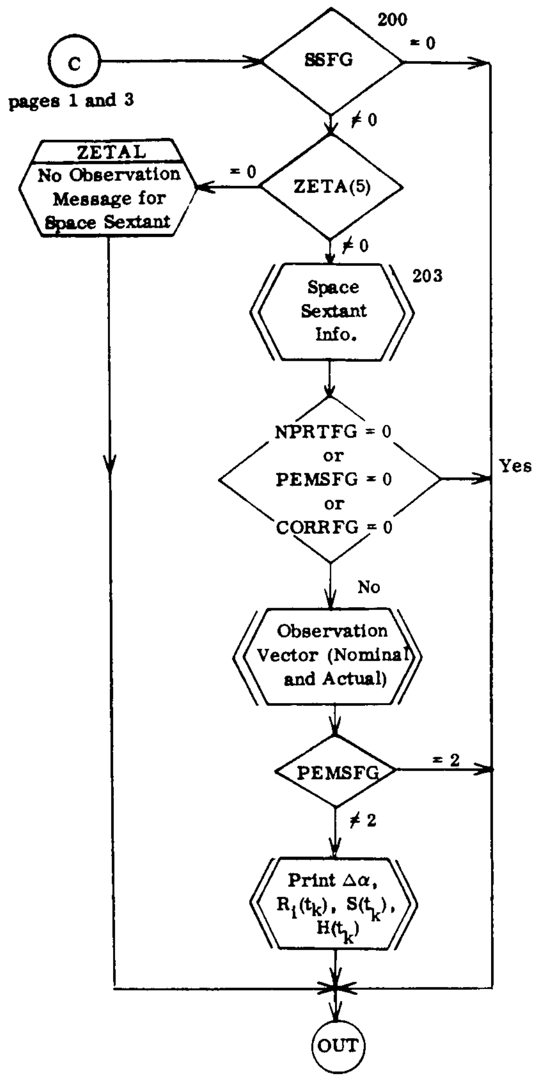
6.4.4.8 EMSPTL - Subroutine



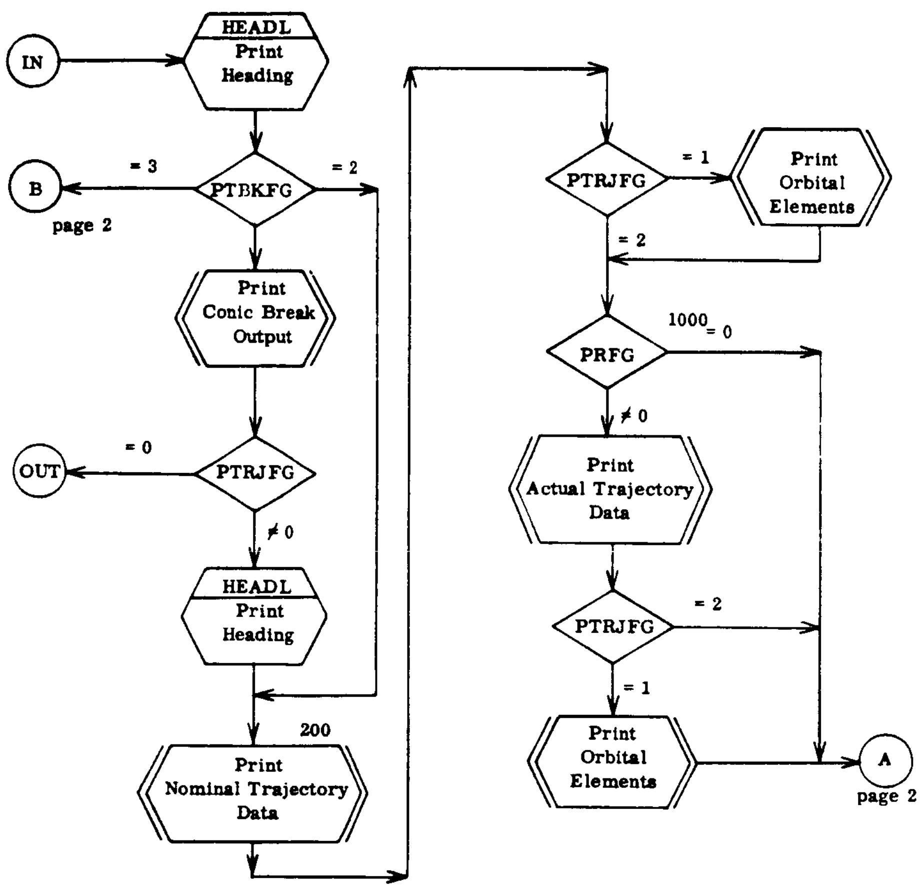
EMSPTL - Subroutine (contd)



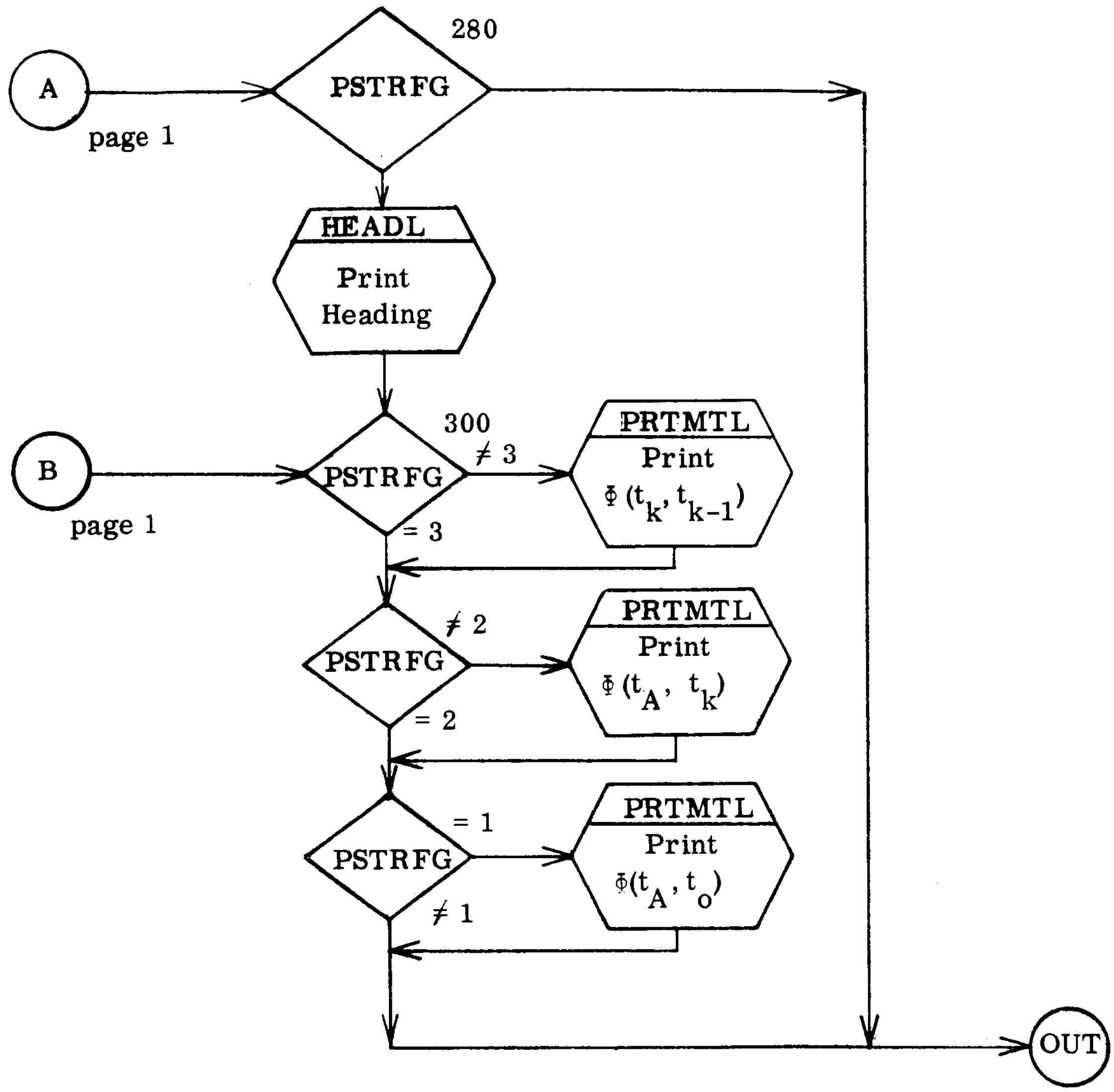
EMPSTL - Subroutine (contd)



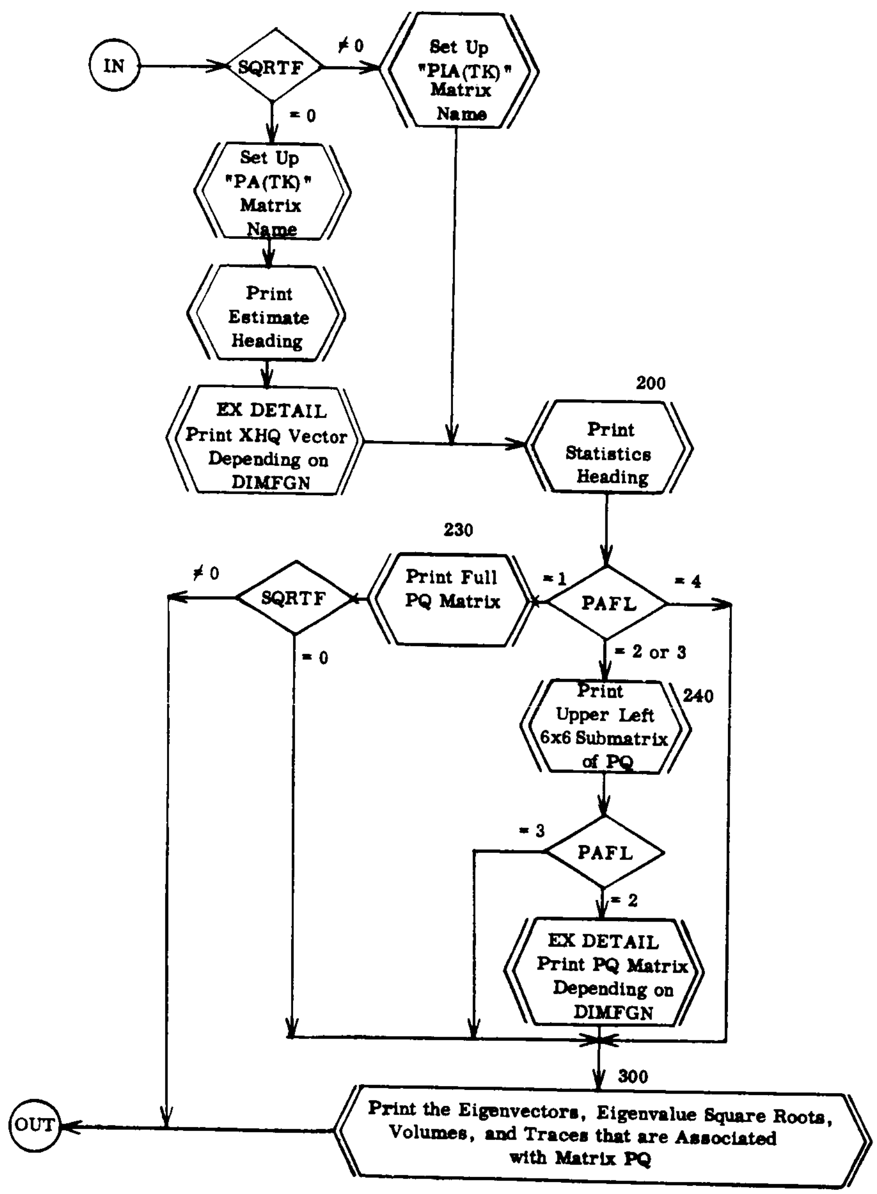
EMSPTL - Subroutine (contd)



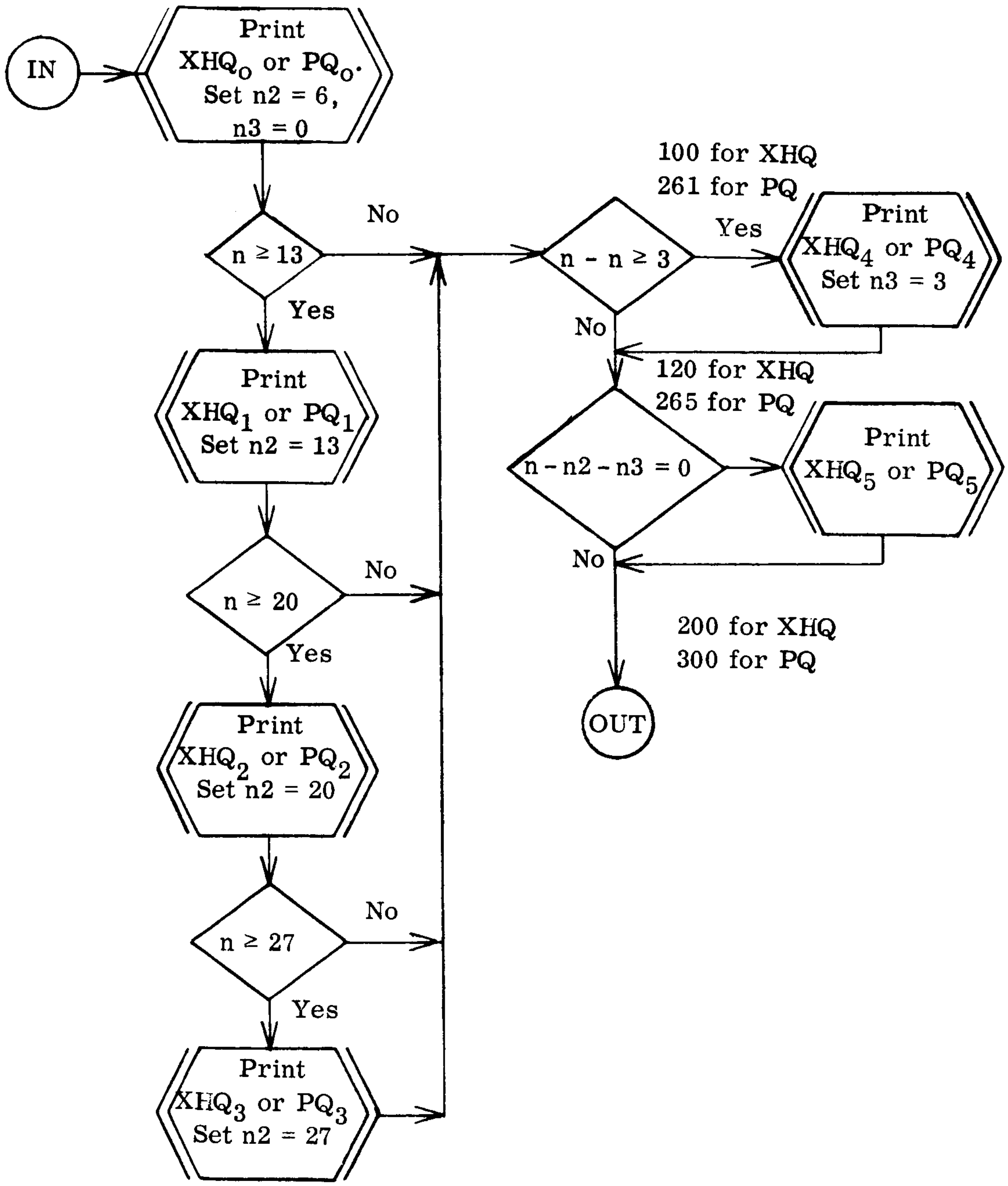
6.4.4.9 BTYSTL - Subroutine



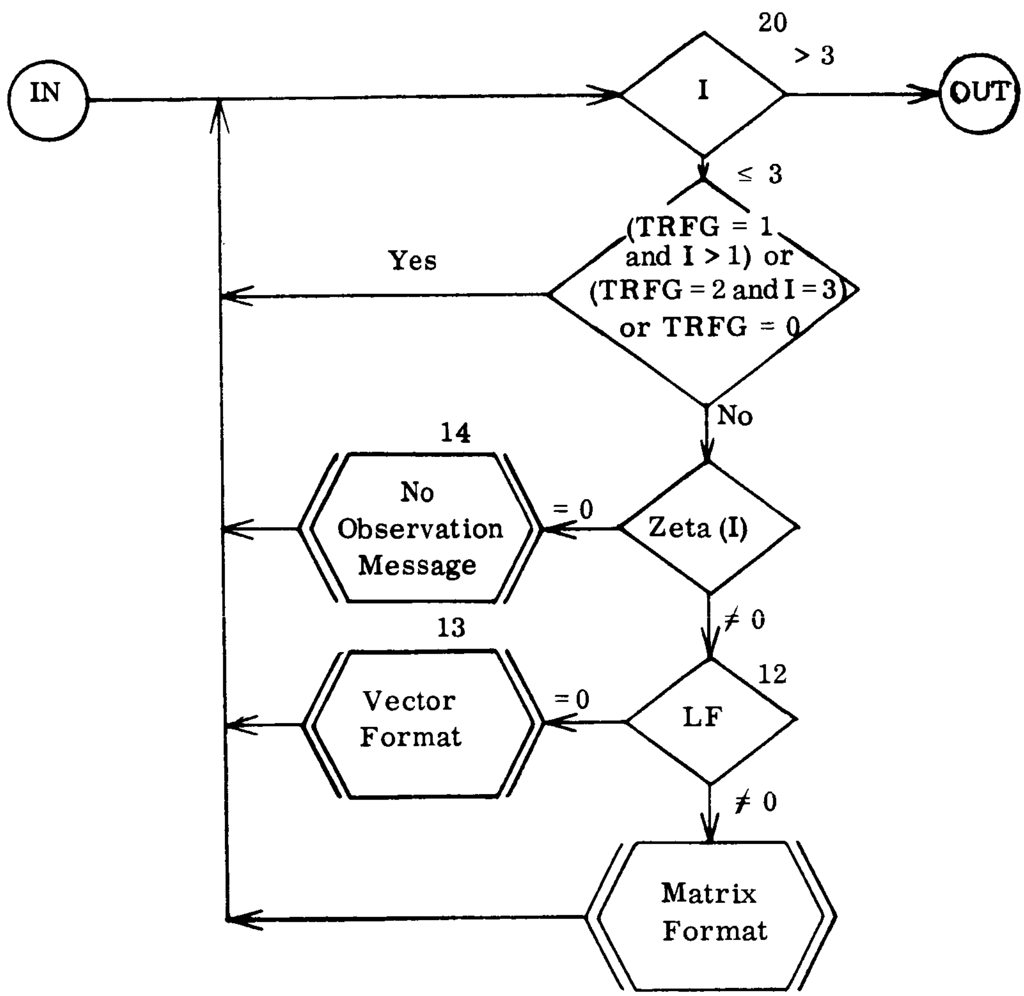
BTYSTL - Subroutine (contd)



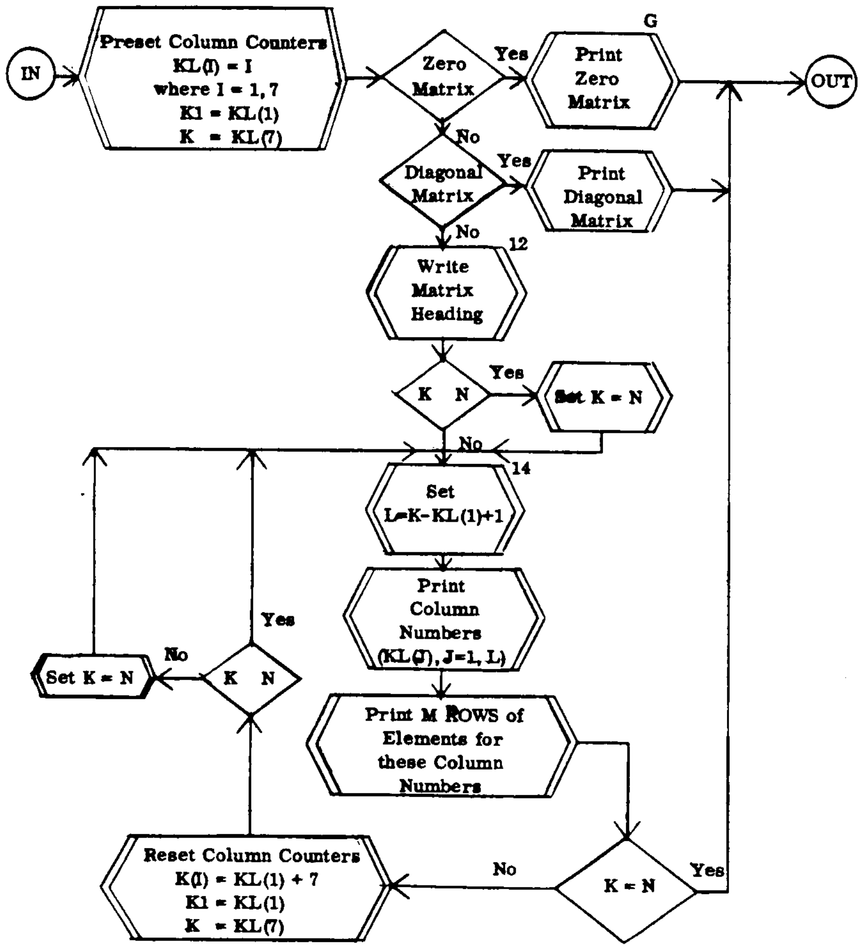
6.4.4.10 EXAUPL - Subroutine



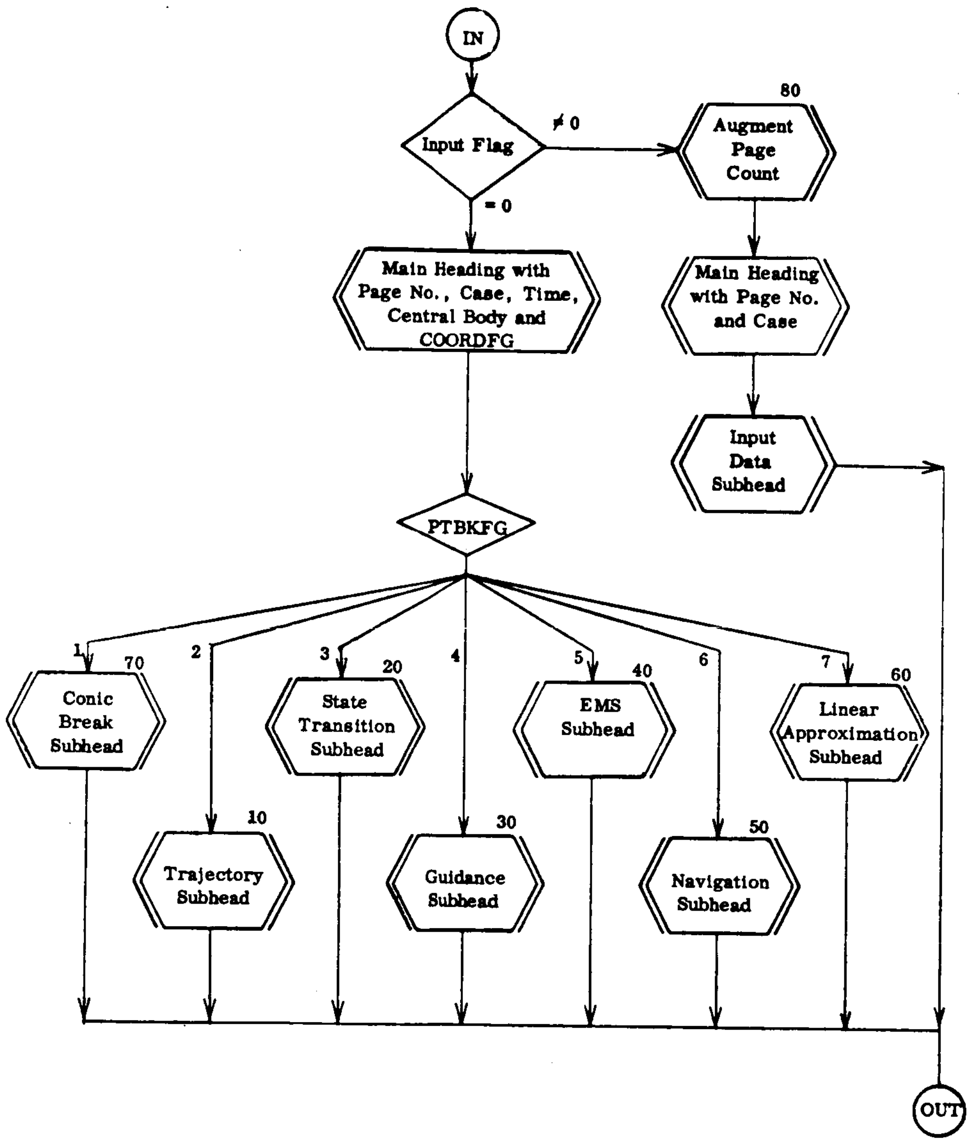
EXAUPL - Ex. Detail



6.4.4.11 ZTINTL - Subroutine



6.4.4.12 PRTMTL - Subroutine



6.4.4.13 HEADL - Subroutine

7.0 REFERENCES

1. Sorenson, H. W., Salomon, Z., Kido, T., "Application of Wiener-Kalman Filter Theory to the Determination of Nearly Circular Orbits", ACRD Report, August 1963.

2. Sorenson, H. W., "Kalman Filtering Techniques", Chapter 5 of Advances in Control Systems, Volume 3, edited by C. T. Leondes, Academic Press, to be published in 1966.

3. Sorenson, H. W., Salomon, Z., "Midcourse Guidance Theory for General Terminal Constraints", ACRD Report No. LAS-0017, August 1964.

4. Salomon, Z., "Explicit Theory of Error Propagation for Elliptic Orbits of Arbitrary Eccentricity ($0 \leq e < 1$)", ACRD Memorandum LA-444, 26 November 1962.

5. Danby, J. M. A., "Matrix Methods in the Calculation and Analysis of Orbits", AIAA Journal, Vol. 2, No. 1, January 1964.

6. "System Capabilities and Development Schedule of the Deep Space Instrumentation Facility 1964-68", J.P.L. Technical Memorandum No. 33-83, April 24, 1964.

7. Battin, R. H., Astronautical Guidance, McGraw-Hill (1964).

8. Moulton, F. R., Celestial Mechanics, The Macmillan Company, 15th Printing 1962.

APPENDIX A

PROGRAM LISTING, PROGRAM 284

The program listing is supplied with the program decks.